

IMU Data Processing MATLAB Functions User Guide

Scott Kresie – Last Updated March 2021

IMPORTANT: Make sure all functions and IMU data are in the same MATLAB path.

Data Processing Functions

importData_fxn.m

Imports IMU data from folder saved on computer. This folder is the output of the X-IO file that is converted by the NGIMU SD Card software which takes the raw X-IO file from the NGIMU and converts it into CSVs.

```
importData_fxn(data_directory, folder_IMU1, folder_IMU2)
```

Arguments:

data_directory is a string-type variable with the directory location of the IMU data folders that were collected by the NGIMU

Example: `data_directory = 'C:\Users\scott\Documents\HPL\IMU\RATA_IMU_DATA\';`

folder_IMU1 is a string-type variable with the folder name for IMU #1 (thigh IMU) data

folder_IMU2 is a string-type variable with the folder name for IMU #2 (shank IMU) data

Example: `folder_IMU1 = 'IMU1_HUMN_11';`

The function creates a structure-type variable with fields for each IMU's accelerometer and gyroscope data as well as each IMU's orientation data in quaternion form as well as time-stamps for power-button presses during the recording session.

Example use:

```
data = importData_fxn(data_directory, folder_IMU1, folder_IMU2)
```

data =

struct with fields:

sensors1: [1×1 struct]

sensors2: [1×1 struct]

quat1: [564844×4 double]

quat2: [504283×4 double]

button1: [90×1 double]

```
data.sensors1 =
```

```
ans =
```

```
struct with fields:
```

```
time: [564844×1 double]
gyroscopeX: [564844×1 double]
gyroscopeY: [564844×1 double]
gyroscopeZ: [564844×1 double]
accelerometerX: [564844×1 double]
accelerometerY: [564844×1 double]
accelerometerZ: [564844×1 double]
magnetometerX: [564844×1 double]
magnetometerY: [564844×1 double]
magnetometerZ: [564844×1 double]
barometer: [564844×1 double]
gyroscopeVector: [564844×3 double]
accelerometerVector: [564844×3 double]
magnetometerVector: [564844×3 double]
```

getDCM_fxn.m

```
getDCM_fxn(data, dcm_button_presses)
```

Arguments:

data is the data structure output by importData_fxn.m

dcm_button_presses is a 1xn vector containing the button press numbers that mark the start and end of each section of data from the DCM calibration protocol. The function assumes the power button IMU #1 (thigh IMU) was pressed.

Example: `dcm_button_presses = [3:6];`
Button press #3: start of straight static standing period
Button press #4: end of straight static standing period
Button press #5: start of straight-legged hip flexion period
Button press #6: end of straight-legged hip flexion period

The function returns a structure-type variable with the IMU to anatomical reference frame DCMs for the thigh and shank IMUs ("DCM1T" and "DCM2S" respectively)

The function also returns structure-type variables that contain the indices corresponding to each button press in the time series for each IMU.

Example Use:

```
DCMs = getDCM_fxn(data, dcm_button_presses);
```

```
DCMs =
```

```
struct with fields:
```

```
DCM1T: [1x1 struct]
DCM2S: [1x1 struct]
thigh_index_stand: [1x1 struct]
thigh_index_flex: [1x1 struct]
shank_index_stand: [1x1 struct]
shank_index_flex: [1x1 struct]
```

Each field has a sub-field 'tr1' which refers to the calibration protocol trial. Usually this will just be 'tr1', but if repeated calibration protocols are taken and the length of dcm_button_presses is longer than 4 (i.e. 8, 12, 16, etc.) then additional sub-fields are generated containing the DCM data and follow the 'tr2', 'tr3', etc. naming scheme.

```
DCMs.DCM1T.tr1
ans =
```

```
-0.1177 -0.9821 0.1472
0.9922 -0.1224 -0.0232
0.0408 0.1433 0.9888
```

```
DCMs.thigh_index_flex.tr1
ans = [ 72524, 76393]
```

NOTE: IMU#1 and IMU#2 data streams are synchronized within the getDCM_fxn.m function.

getposvecs_fxn2.m

```
getposvecs_fxn2(data, DCMs, button_presses)
```

Arguments:

data is the data structure output by importData_fxn.m

DCMs is the data structure output by the getDCM_fxn.m

button_presses is a 1xn vector containing the button press numbers that mark the start and end of each section of data from the position vector calibration protocol. The function assumes the power button IMU #1 (thigh IMU) was pressed.

```
Example: button_presses = [7:10];
Button press #7: start of isolated shank flexion/extension
Button press #8: end of isolated shank flexion/extension
Button press #9: start of isolated thigh flexion/extension
Button press #10: end of isolated thigh flexion/extension
```

The function returns a data structure type variable with fields “posvec1” and “posvec2” referring to the estimated thigh and shank position vectors from the IMU to the estimated knee-joint-center (approximated femoral/tibial origins). Within the posvec1 or posvec2 fields is a subfield indicating the calibration protocol trial. Its default is “tr1”, but if multiple calibration protocols are performed, there will be a tr2, tr3, tr4, etc. sub-field. If multiple calibration protocols are performed the 1xn button_press vector will be of length 8, 12, or 16, etc. This is useful if you are testing the repeatability of the calibration protocol.

Within the subfield posvec1.tr1 is a 1x2 vector containing the x and y components (in meters) of the position vector. The z-component needs to be manually measured at this time.

Example:

```
posvecsA = getposvecs_fxn2(data,DCMs, posvec_button_presses);
```

posvecsA =

struct with fields:

posvec1: [1x1 struct]

posvec2: [1x1 struct]

posvecsA.posvec1.tr1

ans =

[0.0022 -0.2326]

NOTE: IMU#1 and IMU#2 data streams are synchronized within the getposvecs_fxn2.m function.

getRATA_fxn.m

```
getRATA_fxn(data, cutoff_freq, DCMs, posvecs, button_presses)
```

Arguments

data is the data structure output by importData_fxn.m

cutoff_freq is the blanket cutoff frequency (Hz) that is applied to all accelerometer and gyroscope data. If *cutoff_freq* is set to 0, the Yu (1999) filtering method discussed in the thesis is used for each trial defined by a pair of button presses (see *button_presses*). The Yu filtering method uses a sub-section of data (500 data points) about where peak longitudinal shank acceleration occurs which is an approximation of ground contact.

DCMs is the data structure output by the getDCM_fxn.m

posvecs is the data structure output by *getposvecs_fxn2.m*

button_presses is a 1xn vector containing the button press numbers that mark the start and end of each section of data from the RATA data collection phase. For mechanical system testing, these button presses represent just prior to and just after the system was dropped and RATA occurred. Multiple drops (trials) could occur for each IMU recording session. For human subject testing, these button presses represent the start and end of a movement trial (button press before a jumpstop and button press after completion of the jumpstop is one trial).

The function returns a data-structure type variable with fields that are also structures. The main variables that are used in further data processing are RATA, thigh_acc_kjc, and shank_acc_kjc.

Example:

```
IMU_RATA = getRATA_fxn(data, cutoff_freq, DCMs, posvecs, button_presses)
```

```
IMU_RATA =
```

struct with fields:

```
RATA: [1x1 struct]
shank_gyr: [1x1 struct]
shank_alpha: [1x1 struct]
shank_acc_kjc: [1x1 struct]
thigh_gyr: [1x1 struct]
thigh_alpha: [1x1 struct]
thigh_acc_kjc: [1x1 struct]
thigh_acc_kjc_tframe: [1x1 struct]
shank_acc_kjc_tframe: [1x1 struct]
max_RATA_index: [1x1 struct]
cutoff_freqs: [1x1 struct]
IMU1_raw: [1x1 struct]
IMU2_raw: [1x1 struct]
IMU1_filt: [1x1 struct]
IMU2_filt: [1x1 struct]
```

Each structure has fields separating data for each trial with trial names 'tr1', 'tr2', etc. The definitions for each field are as follows:

RATA – RATA for each trial. No phase shifting between the tibial and femoral origin acceleration signals

shank_gyr – Nx3 matrix, X,Y, and Z gyroscope (angular velocity) data in the shank anatomical reference frame.

shank_alpha – Nx3 matrix, X,Y, and Z angular acceleration data in the shank anatomical reference frame.

shank_acc_kjc – Nx3 matrix, X,Y, and Z acceleration data of the tibial origin in the shank anatomical reference frame.

thigh_gyr – Nx3 matrix, X,Y, and Z gyroscope (angular velocity) data in the thigh anatomical reference frame.

thigh_alpha – Nx3 matrix. X,Y, and Z angular acceleration data in the thigh anatomical reference frame.

thigh_acc_kjc – Nx3 matrix. X,Y, and Z femoral origin acceleration in the tibial anatomical reference frame.

thigh_acc_kjc_tframe – Nx3 matrix. X, Y, and Z acceleration of the femoral origin in the thigh/femoral anatomical reference frame

shank_acc_kjc_tframe – Nx3 matrix. X, Y, and Z acceleration of the tibial origin in the thigh/femoral anatomical reference frame.

max_RATA_index – index where max RATA occurs for a particular trial. RATA is calculated without any shifting between the tibial and femoral origin acceleration signals.

cutoff_freqs – cutoff frequencies (Hz) for each axis for each accelerometer and gyroscope from both IMUs. Will be constant if *cutoff_freq* variable was set as a constant, but will vary if *cutoff_freq* was set to 0 (Yu filter method).

IMU1_raw – Raw (unfiltered) X,Y,Z acceleration and X,Y,Z angular velocity of IMU#1's data in the thigh anatomical reference frame

IMU2_raw – Raw (unfiltered) X,Y, and Z acceleration and X,Y, and Z angular velocity of IMU#2's data in the shank anatomical reference frame.

IMU1_filt -- Filtered, X,Y, and Z acceleration, X,Y, and Z angular velocity and X,Y, and Z angular acceleration of IMU#1's data in the thigh anatomical reference frame.

IMU2_filt -- Filtered, X,Y, and Z acceleration, X,Y, and Z angular velocity and X,Y, and Z angular acceleration of IMU #2's data in the thigh anatomical reference frame.

[mocap_processing_fxn.m](#)

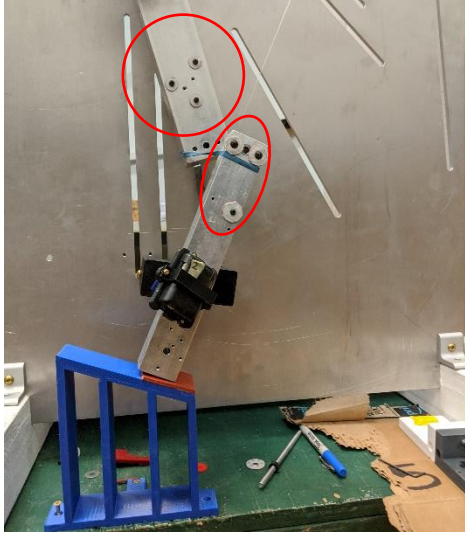
`mocap_processing_fxn(mocap_ascii_filename, marker_config)`

Arguments

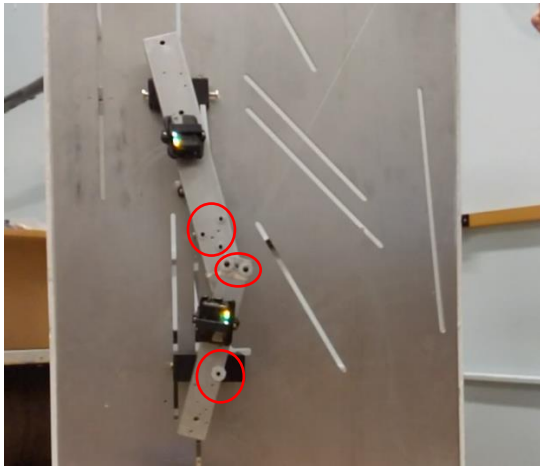
mocap_ascii_filename is the ASCII filename exported out of MAXTRAQ motion capture software. It should be a (.mqa) file.

marker_config is a numerical identifier that refers to the configuration/placement of the motion capture markers on the mechanical test rig. Currently there are three different configurations that have been used.

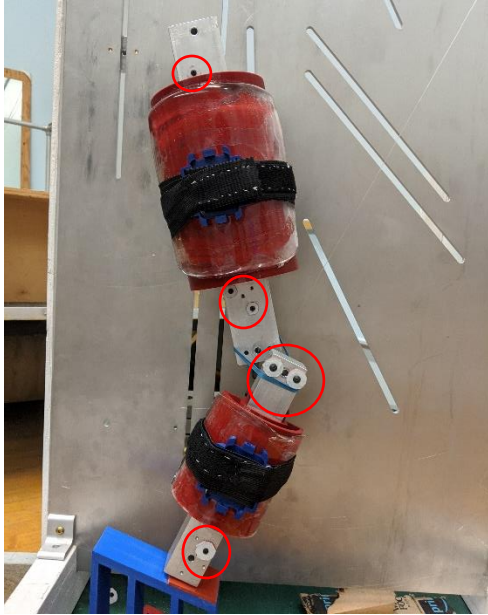
marker_config = 1 -> The IMUs were in either the BBBB or BBCB configurations on the thigh and shank. (See Chapter 3 in Thesis for more info on IMU configuration)



marker_config = 2 -> The IMUs were in the BBAB configuration.



marker_config = 3 -> The IMUs were strapped to the soft-tissue ballistic gel.



The function returns a data-structure that has fields for each trial (i.e. 'tr22'). The most important variables are RATA_mocap, acc_kjc_shank, and acc_kjc_thigh. The remaining output variables are supplementary for optional additional analysis.

Example:

```
MOCAP_RATA = mocap_processing_fxn(mocap_ascii_filename, marker_config)
```

```
MOCAP_RATA.tr22
```

```
ans =
```

struct with fields:

```
RATA_mocap: [282×1 double]
ATT_mocap: [282×1 double]
ATV_mocap: [282×1 double]
acc_kjc_shank: [282×2 double]
acc_kjc_thigh: [282×2 double]
p_kjc_shank: [282×1 double]
p_kjc_thigh: [282×1 double]
fc_t_angle: 12.5000
fc_s_angle: 19
fc_t_x: 9
fc_t_y: 25
fc_s_x: 20
fc_s_y: 24.5000
vel_t1: [282×2 double]
vel_s1: [282×2 double]
acc_t1: [282×3 double]
acc_s1: [282×3 double]
```


omega_s_mocap: [282×1 double]
omega_t_mocap: [282×1 double]
alpha_s_mocap: [282×1 double]
alpha_t_mocap: [282×1 double]
pix_dist_ratio_thigh: 5.5146e-04
pix_dist_ratio_shank: 5.2763e-04
thigh_raw: [282×2 double]
shank_raw: [282×2 double]
thigh_filt: [282×2 double]
shank_filt: [282×2 double]
theta_s_raw: [282×1 double]
theta_s_filt: [282×1 double]
theta_t_raw: [282×1 double]
theta_t_filt: [282×1 double]
acc_kjc_thigh_xy: [282×2 double]
acc_kjc_shank_xy: [282×2 double]

The definitions for each of these fields are as follows:

RATA_mocap: RATA calculated using the motion capture markers
ATT_mocap: anterior tibial translation calculated directly from motion capture marker positional data.
ATV_mocap: first derivative of ATT_mocap.
acc_kjc_shank: tibial origin acceleration in the anterior tibial direction
acc_kjc_thigh: femoral origin acceleration in the anterior tibial direction
p_kjc_shank: 1-D positional data of the tibial origin in the anterior tibial direction.
p_kjc_thigh: 1-D positional data of the femoral origin in the anterior tibial direction.
fc_t_angle: low-pass cutoff frequency for thigh sagittal-plane angle data
fc_s_angle: low-pass cutoff frequency for the shank sagittal-plane angle data
fc_t_x: low-pass cutoff frequency for the x-position thigh marker data
fc_t_y: low-pass cutoff frequency for the y-position thigh marker data
fc_s_x: low-pass cutoff frequency for the x-position shank marker data
fc_s_y: low-pass cutoff frequency for the y-position shank marker data
vel_t1: first-derivative of x and y position data of a single thigh marker. [Nx2 double], (x,y)
vel_s1: first-derivative of x and y position data of a single shank marker. [Nx2 double], (x,y)
acc_t1: second-derivative of x and y position data of single thigh marker. [Nx2 double], (x,y)
acc_s1: second-derivative of x and y position data of a single shank marker. [Nx2 double], (x,y)
omega_s_mocap: first-derivative of the shank angle data. The angle data was derived from marker position data.
omega_t_mocap: first-derivative of the thigh angle data.
alpha_s_mocap: second-derivative of the shank angle data.
alpha_t_mocap: second-derivative of the thigh angle data.
pix_dist_ratio_thigh: ratio between pixels and meters for the thigh markers (meters/pixel)
pix_dist_ratio_shank: ratio between pixels and meters for the shank markers (meters/pixel)
thigh_raw: raw position data for a marker on the thigh segment (x-y coordinate system, not anatomical)
shank_raw: raw position data for a marker on the shank segment (x-y coordinate system, not anatomical)
thigh_filt: low-pass Butterworth filtered *thigh_raw* position data

shank_filt: low-pass Butterworth filtered *shank_raw* position data.
theta_s_raw: raw shank angle data
theta_s_filt: low-pass Butterworth filtered shank angle data
theta_t_raw: raw thigh angle data
theta_t_filt: low-pass Butterworth filtered thigh angle data
acc_kjc_thigh_xy: femoral origin acceleration in the x-y coordinate system, not in the anatomical coordinate system.
acc_kjc_shank_xy: tibial origin acceleration in the x-y coordinate system, not in the anatomical coordinate system.

Data Analysis Scripts

Three main data analysis scripts were used to analyze data from the mechanical test rig, mechanical test rig plus soft-tissue, and human subject test conditions. The scripts are based largely on the use of the functions described previously. The following should be a guide to users on how to appropriately run these scripts by adjusting variables within them which depend on how the data was collected.

[RATA_PROTOCOL_MECHTestRig_v2.m](#)

There are several lines of code in this script that may require adjustment (uncommenting/commenting lines, for example) by the user depending on how the data being analyzed was collected. The script outputs key variables for each drop-trial like RMSE between MOCAP and IMU derived RATA, peak RATA from each method, cross-correlation coefficient between measurement methods, and difference in peak RATA. These are output under the 'a' variable. See Lines 444-454.

Example:

```
s(:,1) = transpose(struct2mat(shift_amt.shank)); %soft-tissue shift
s(:,2) = transpose(struct2mat(shift_amt.thigh)); %soft-tissue shift
a(:,1) = transpose(struct2mat(pk_IMU_RATA)); %peak RATA (IMU-derived)
a(:,2) = transpose(struct2mat(pk_MOCAP_RATA)); %peak RATA (mocap-derived)
a(:,3) = transpose(struct2mat(pk_shiftRATA)); %peak RATA(IMU-derived, with shift
applied, (soft-tissue only))
a(:,4) = a(:,1)-a(:,2); %pk IMU RATA - pk MOCAP RATA
a(:,5) = a(:,3)-a(:,2); %pk SHIFT IMU RATA (thigh/shank acc signal relative shift
applied) - pk MOCAP RATA
a(:,6) = transpose(struct2mat(RMSE)); %RMSE between RATA signals from two measurement
methods
a(:,7) = transpose(struct2mat(RMSE_shift)); %RMSE calculated with relative shift
applied, (soft-tissue only)
a(:,8) = transpose(struct2mat(R_struct)); %xcorr coeff
a(:,9) = transpose(struct2mat(R_shiftRATA)); %xcorr coeff with shift appplied (soft-
tissue only)
```

Code Adjustments based on IMU Configuration

DCMs

This data analysis script uses the DCMs and position vectors calculated during repeated calibration experiments that evaluated the repeatability of the protocol using the mechanical test rig. The

appropriate DCMs and position vectors corresponding to the correct IMU configuration needs to be specified in the script.

Example:

Lines 9-17 & 25-33 contain uncommented/commented variables that will define which DCM configuration is used. The uncommented foldernames must match for IMU #1 INFO lines and for the IMU #2 INFO lines, but they do not have to all match.

```
%% (IMU #1 INFO)
folder_IMU1 = 'IMU1_MECH_02_BBBB'; %foldername for IMU #1 data where DCM for IMU#1 is
accessed
folder_IMU2 = 'IMU2_MECH_02_BBBB'; % foldername for IMU #2 data where DCM for IMU#2 is
accessed

% folder_IMU1 = 'IMU1_MECH_01_ABAB'; %foldername for IMU #1 data
% folder_IMU2 = 'IMU2_MECH_01_ABAB'; % foldername for IMU #2 data

% folder_IMU1 = 'IMU1_MECH_04_CBCB'; %foldername for IMU #1 data
% folder_IMU2 = 'IMU2_MECH_04_CBCB'; % foldername for IMU #2 data

%% (IMU #2 INFO)
folder_IMU1 = 'IMU1_MECH_02_BBBB'; %foldername for IMU #1 data
folder_IMU2 = 'IMU2_MECH_02_BBBB'; % foldername for IMU #2 data

% folder_IMU1 = 'IMU1_MECH_01_ABAB'; %foldername for IMU #1 data
% folder_IMU2 = 'IMU2_MECH_01_ABAB'; % foldername for IMU #2 data

% folder_IMU1 = 'IMU1_MECH_04_CBCB'; %foldername for IMU #1 data
% folder_IMU2 = 'IMU2_MECH_04_CBCB'; % foldername for IMU #2 data
```

The code above is appropriate if the mechanical test rig measured RATA had IMUs in the 'BBBB' configuration. However, by commenting out folder_IMU1 and folder_IMU2 = 'IMU1_MECH_02_BBBB' in the IMU #2 INFO section of code and un-commenting the lines with folder_IMU1 = 'IMU1_MECH_01_ABAB', and folder_IMU2 = 'IMU2_MECH_01_ABAB', data can be analyzed for IMUs placed in the BBAB configuration. IMU #1 is in the BB configuration and IMU#2 is in the AB configuration.

Position Vectors

This process of commenting/un-commenting specific lines depending on the IMU configuration is also used when pulling the position vector information from previously performed position vector calibration repeatability experiments.

See lines 45-58 & 68-77.

Example:

```
%% IMU 1 INFO
%% PosVecs (USER INPUT, button presses if more than 1 trial collected (optional) )

data_directory = 'C:\Users\scott\Documents\HPL\IMU\POS_VEC_DATA\'; %location of IMU
data
clear folder_IMU1
clear folder_IMU2
folder_IMU1 = 'IMU1_MECH_16_BBBB'; %foldername for IMU #1 data
folder_IMU2 = 'IMU2_MECH_16_BBBB'; % foldername for IMU #2 data
```

```

% folder_IMU1 = 'IMU1_MECH_14_ABAB'; %foldername for IMU #1 data (BBAB config)
% folder_IMU2 = 'IMU2_MECH_14_ABAB'; % foldername for IMU #2 data (BBAB config)

% folder_IMU1 = 'IMU1_MECH_11_CBCB'; %foldername for IMU #1 data (BBCB config)
% folder_IMU2 = 'IMU2_MECH_11_CBCB'; % foldername for IMU #2 data (BBCB config)

%% IMU 2 INFO
%% POSVEC 2
folder_IMU1 = 'IMU1_MECH_16_BBBB'; %foldername for IMU #1 data (BBBB)
folder_IMU2 = 'IMU2_MECH_16_BBBB'; % foldername for IMU #2 data (BBBB)
%
% folder_IMU1 = 'IMU1_MECH_14_ABAB'; %foldername for IMU #1 data (BBAB config)
% folder_IMU2 = 'IMU2_MECH_14_ABAB'; % foldername for IMU #2 data (BBAB config)

% folder_IMU1 = 'IMU1_MECH_11_CBCB'; %foldername for IMU #1 data (BBCB config)
% folder_IMU2 = 'IMU2_MECH_11_CBCB'; % foldername for IMU #2 data (BBCB config)

```

MOCAP Data

Motion capture data was collected with IMUs in different configurations. The script needs to be adjusted to reflect the specific configuration that is being analyzed. Commenting/un-commenting the lines of code corresponding to the appropriate IMU configuration can be done in lines 93-100. These lines contain parts of the MOCAP filename is associated with each configuration.

Example:

```

data_directory = 'C:\Users\scott\Documents\HPL\IMU\MAXTRAQ\'; %location of
IMU data
%% BBBB folder numbers
mocap_num = {'22_1'; '23_2';
'24_3'; '25_4'; '26_5'; '27_1'; '28_2'; '29_3'; '30_4'; '31_5'; '50_1'; '51_2'; '52_3';
'53_4'; '54_5'; '65_1'; '66_2'; '67_3'; '68_4'; '69_5'; '70_6'; '71_7'; '72_8'; '73_9';
'74_10';}; % BBBB config trials

%% BBAB folder numbers
% mocap_num = {'37_1'; '38_2'; '39_3'; '40_4'; '41_5';
'42_1'; '43_2'; '44_3'; '45_4'; '46_5'};

%% BBCB folder numbers
% mocap_num = {'32_1'; '33_2'; '34_3'; '35_4'; '36_5'};

```

IMU RATA Data

The script also may need to be adjusted so that the correct IMU data filenames corresponding to the appropriate IMU configuration is accessed.

See Lines 113-116 & Lines 173&174

Example:

```

%% USER INPUT HERE!! %%%
imu_num =
{'22'; '23'; '24'; '25'; '26'; '27'; '28'; '29'; '30'; '31'; '50'; '51'; '52'; '53'; '54';
'65'; '66'; '67'; '68'; '69'; '70'; '71'; '72'; '73'; '74'}; %BBBB config
%imu_num = {'37'; '38'; '39'; '40'; '41'; '42'; '43'; '44'; '45'; '46'}; %BBAB config
%imu_num = {'32'; '33'; '34'; '35'; '36'}; % BBCB config

```

Lines 173&174

```
folder_IMU1 = ['IMU1_MECH_',imu_num{n},'_BBBB']; %% USER INPUT HERE!! %%%foldername
for IMU #1 data
folder_IMU2 = ['IMU2_MECH_',imu_num{n},'_BBBB']; %% USER INPUT HERE!! %%% foldername
for IMU #2 data
```

The string 'BBBB' may need to be changed to match with the folder name of RATA IMU data being analyzed which could be 'BBAB', 'BBCB', etc.

Other Code Adjustments

There is a section of code in the script which can be used to find the optimal cutoff frequency for a low-pass Butterworth filter. This is a "blanket" cutoff frequency that is applied to all gyro and accelerometer data of both IMUs. The optimal frequency for each trial is found by minimizing the difference between the MOCAP and IMU derived peak RATA for each trial.

See Line 126:

Example:

```
%% OPTIMAL FREQ CODE
data = importData_fxn(data_directory,folder_IMU1,folder_IMU2);
freqs = [18:35]; %%USER INPUT, range of possible optimal frequencies to
search within.
```

The above code will search in the 18-35Hz range for the optimal frequency.

The output variable "peaks" is a structure variable with fields that show the cutoff frequencies that correspond to 10%, 15% and 20% differences between the MOCAP and IMU derived peak RATA.

Example:

```
peaks.twenty_pct_range.(trial_fieldname) = freqs(abs(peaks.diff.(trial_fieldname)) ./
peaks.mocap.(trial_fieldname) <0.2); %cutoff freqs that result in less than 20%
difference between imu and mocap derived RATA
peaks.fifteen_pct_range.(trial_fieldname) = freqs(abs(peaks.diff.(trial_fieldname)) ./
peaks.mocap.(trial_fieldname) <0.15); % '' less than 15% diff between imu and mocap
derived RATA.
peaks.ten_pct_range.(trial_fieldname) = freqs( abs(peaks.diff.(trial_fieldname)) ./
peaks.mocap.(trial_fieldname) < 0.1); % '' less than 10% ''
```

SEE LINE 181. This is where the getRATA_fxn.m function is called and where the blanket cutoff frequency is set or if the Yu filtering protocol is used.

Example:

```
IMU_RATA.(trial_fieldname) = getRATA_fxn(data,20,DCMs,posvecs,rata_button_presses);
%getRATA_fxn(sessionData,cutoff filter freq, DCMs, posvecs, button presses in session
for RATA trials);
```

The above code is set for a 20Hz cutoff frequency, but if the second argument was 0, the Yu filtering protocol would be used.

RATA_softtissue_analysis.m

There are several lines of code in this script that may require adjustment (uncommenting/commenting lines, for example) by the user depending on how the data being analyzed was collected. The script outputs key variables for each drop-trial like RMSE between MOCAP and IMU derived RATA, peak RATA from each method, cross-correlation coefficient between measurement methods, and difference in peak RATA. These key variables are calculated for when the phase lag between tibial and femoral origin acceleration signals are and are not accounted for. These are output under the 'a' variable. See Lines 267-277.

Example:

```
s(:,1) = transpose(struct2mat(shift_amt.shank));
s(:,2) = transpose(struct2mat(shift_amt.thigh));
a(:,1) = transpose(struct2mat(pk_IMU_RATA));
a(:,2) = transpose(struct2mat(pk_MOCAP_RATA));
a(:,3) = transpose(struct2mat(pk_newRATA));
a(:,4) = a(:,1)-a(:,2); %pk IMU RATA - pk MOCAP RATA
a(:,5) = a(:,3)-a(:,2); %pk SHIFT IMU RATA (thigh/shank acc signal relative
shift applied) - pk MOCAP RATA
a(:,6) = transpose(struct2mat(RMSE_noshift));
a(:,7) = transpose(struct2mat(RMSE_shift));
a(:,8) = transpose(struct2mat(R_i_struct));
a(:,9) = transpose(struct2mat(R_shiftRATA));
```

Code Adjustments

These adjustments need to be made for each IMU recording

IMU Data Folder names:

The folder names where the data from each IMU is stored needs to be updated before the script is run.

See Lines 8-9.

Example:

```
folder_IMU1 = 'IMU1_MECH_79'; %foldername for IMU #1 data
folder_IMU2 = 'IMU2_MECH_79'; % foldername for IMU #2 data
```

Button Presses:

Depending on how many trials of RATA data were collected the *rata_button_presses* variable will need to be adjusted.

See Line 23.

Example:

```
%% USER INPUT
rata_button_presses = [11:48, 51:52]; %% USER INPUT HERE!! %%%
```

In the case of the code snippet above, there were a total of 20 trials collected during the same IMU recording session. The motion capture data was not collected for button presses #49-#50 so that IMU data was not analyzed.

IMU Data Cutoff Filter Frequency

The cutoff filter frequency needs to be specified for each recording session.
See Line 76.

Example:

```
cut_f = 20; % cutoff frequency
IMU_RATA = getRATA_fxn(data, cut_f, DCMs, posvecs, rata_button_presses);
%getRATA_fxn(sessionData, cutoff filter freq, DCMs, posvecs, button presses in
session for RATA trials);
```

In the code snippet above a blanket 20Hz cutoff frequency would be applied to all the gyro and accelerometer data of both IMUs. Had *cut_f* been set to 0, the Yu (1999) filter protocol would have been used.

[human_subject_tests_Dave_activity.m](#)

Code Adjustments

IMU Data Folder names:

The folder names where the data from each IMU is stored needs to be updated before the script is run.

See Lines 6-7.

Example:

```
folder_IMU1 = 'IMU1_HUMN_11'; %foldername for IMU #1 data
folder_IMU2 = 'IMU2_HUMN_11'; % foldername for IMU #2 data
```

Button Presses

The button presses which indicate the start and end of the DCM and position vector calibration data collection as well as the start and end of each movement trial need to be defined.

See Lines 12,36, and 45

Example:

```
dcm_button_presses = [3:6]; %% user input here!!%%
posvec_button_presses = [7:10]; %%user input here!!%%
```

```
rata_button_presses = [11:12,15:24];
```

User Interaction With Script Outputs

The analysis of the data collected from human subjects is not currently 100% automated. There is some required user interaction. The user manually selects a snippet of data from the larger section of data that was collected between button presses. This snippet of data is about 50 ms before and 250ms after ground contact of a jumpstop landing, for example. Initial pilot testing used the max longitudinal acceleration of the tibial origin as an approximation of ground contact and data snippets were selected based on this approximation. The user will just need to click once at the rising edge of the peak longitudinal acceleration curves. These click locations are represented by the vertical dashed green lines in Appendix A4 of the thesis (also below in this document).

The user must also identify and match the corresponding peaks and valleys from the tibial and femoral origin acceleration data which are plotted by the script. A separate plot is generated for each pair of rata_button_presses (if rata_button_presses = [11:14, 20:23] 4 plots with the tibial and femoral origin acceleration would be plotted).

Examples of this process are discussed in Appendix A4 in the thesis, but have also been copied here for reference.

Appendix A4: Human subject data processing for RATA – Inflection points and initial ground contact

Inflection Point Pairing

During human subject testing soft-tissue differences experienced by IMU #1 and IMU #2, worn on the thigh and shank respectively, cause phase lag between the tibial and femoral origin acceleration signals. To address the phase lag, peak RATA values were calculated using paired inflection points between the anterior acceleration signal of the tibial and femoral origin that produced the maximum positive difference (or minimum negative difference) of anterior acceleration. The underlying assumption is that the two acceleration signals have similar shape and the peaks and valleys (inflection points) of one signal correspond with peaks and valleys of the other signal.

The inflection point pairing protocol followed a few simple rules:

1. Peaks were paired with peaks and valleys with valleys.

2. Only inflection points in the first 100 ms after initial contact were considered. At least one inflection point in the pair needed to occur in the first 100ms for the pair to be considered (Figure 38A)
3. Some acceleration signals had additional inflection points that could not be paired. These were points were ignored (Figure 38C)

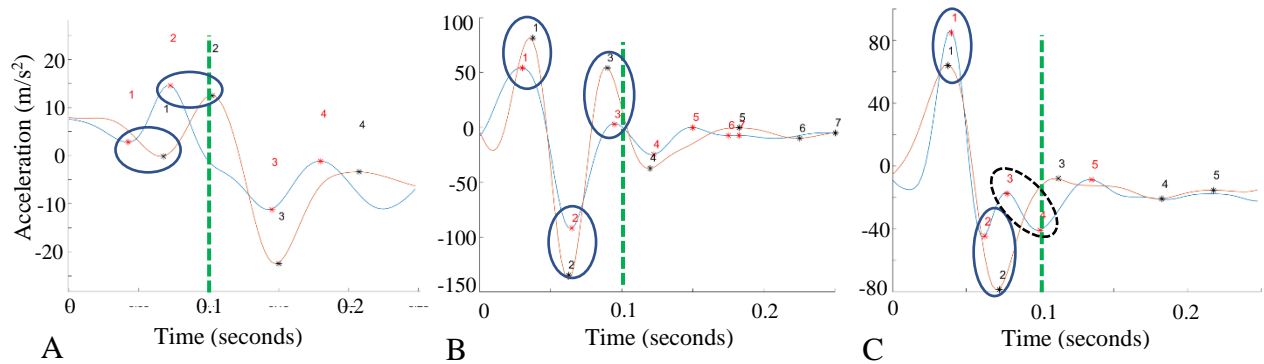


Figure 38 Examples of tibial (blue) and femoral (orange) origin anterior acceleration (in tibial frame) for walking (A), jump-stop (B) and run and cut (C) activities. Vertical dashed lines indicate the 100ms after initial contact cutoff time. Inflection points are marked with red and black asterisks and numbered for each signal. Solid ellipses mark which inflection points were paired. The dashed ellipse (C) indicates additional inflection points in the tibial origin signal which did not have corresponding pairs in the femoral origin signal and were ignored from the analysis.

Initial Ground Contact

The time window of interest for RATA was conservatively set at 0-100ms after initial ground contact based on previous studies regarding the timing of ACL loading and injury [10], [27].

Angular velocity of the shank and thigh and acceleration of the tibial origin in the tibial frame were used to define the time of initial contact. For the vertical jump activity, two initial contact time points were identified: one at toe-contact and one at heel-contact (Figure 39).



Figure 39 Screen-capture of a vertical jump trial indicating two significant contact events, toe and heel-contact. Both contacts induced an observable longitudinal (Y) acceleration peak and peak RATA was calculated relative to each contact event.

The definitions for initial ground contact for each activity were:

Walking: An identifiable valley in the shank's angular velocity signal indicated heel-strike during walking (see Figure 40). This is based on the method used by Bötzel et al. (2016) [55].

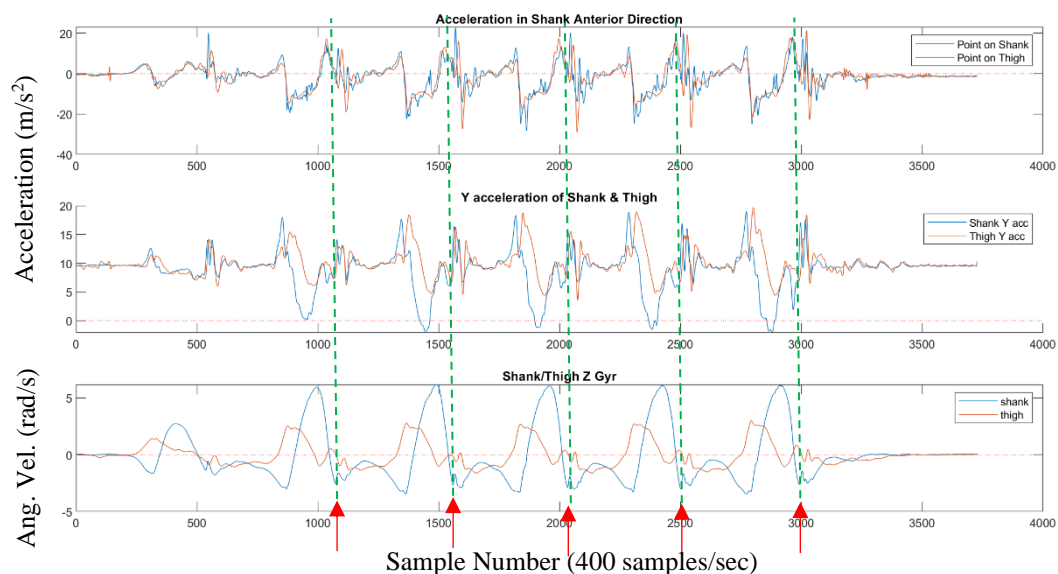


Figure 40 Five heel-strike initial contacts (red arrows) during a walking trial are identified using the shank angular velocity signal (bottom plot). There are also noticeable spikes in the longitudinal (Y) acceleration that occur just after identified heel-strike (middle plot) which helps to confirm these as heel-strikes.

Jogging: Similar to walking, there is an identifiable valley in the shank angular velocity signal near initial ground contact. There is also an identifiable rising edge and peak in the longitudinal (+Y) acceleration for the tibial origin. The start of the rising edge was used as the time of initial ground contact (see Figure 41).

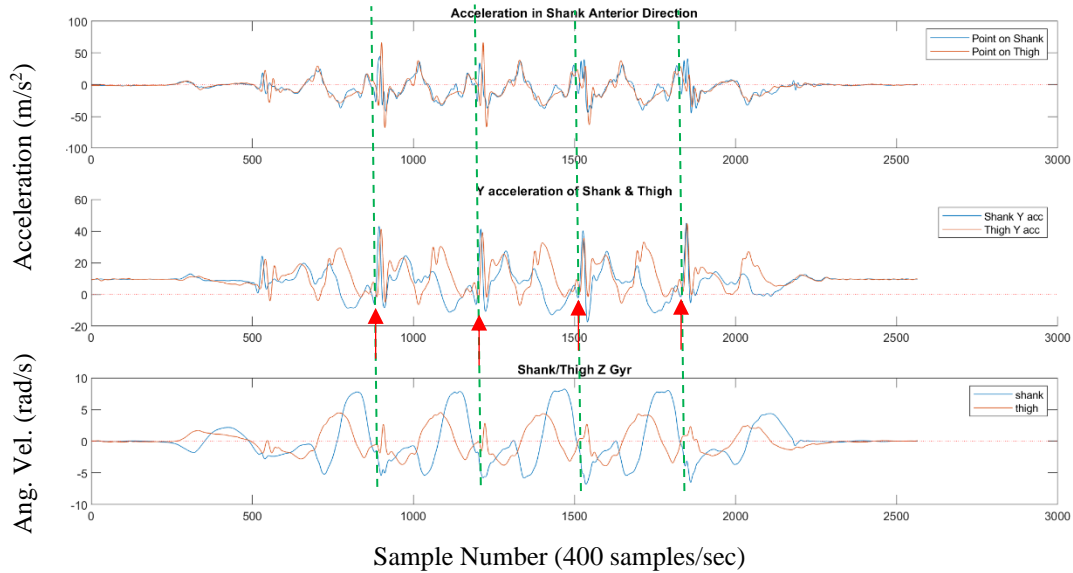


Figure 41 Four initial ground contacts (red-arrows) during a jogging trial. The angular velocity signal is used to identify the data near initial contact and the rising edge of the longitudinal tibial origin acceleration (blue signal) is used to define initial contact.

Vertical Jump: During a vertical jump there is an observable period just prior to initial contact of free-fall where the tibial origin longitudinal (+Y) acceleration is approximately 0 m/s^2 . This was observed because the shank is nearly aligned with gravity during this free-fall period and IMU-derived free-fall data measures 0 m/s^2 for gravitational acceleration rather than -9.81 m/s^2 due to the physical measurement mechanism of the accelerometer in the IMU. The rising edge of the longitudinal (+Y) tibial origin acceleration just after the free-fall period was used to define initial toe-contact. A second rising edge of the longitudinal tibial origin acceleration after the first peak caused by toe-contact was used to define initial heel-contact (see Figure 42).

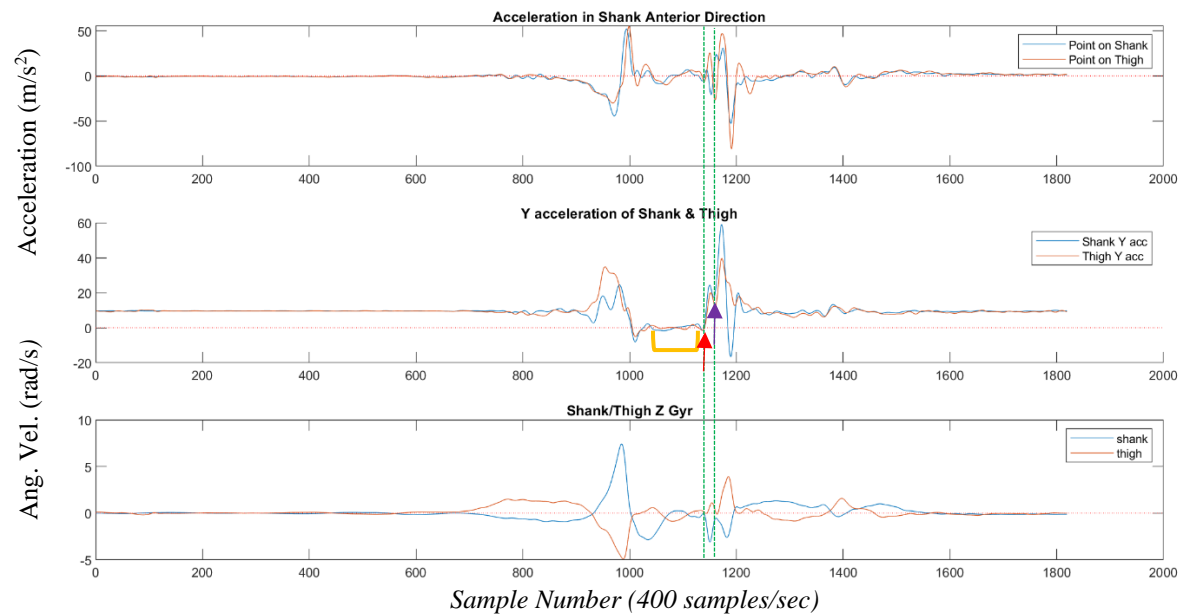


Figure 42 Example of vertical jump free-fall (yellow bracket) followed by identified toe (red arrow) and heel-contact (purple arrow).

Jump-stop: The initial contact for the jump-stop was defined in a similar way to the vertical jump. The free-fall period was used to identify the data near the initial contact event, but rather than two contact events (toe and heel-contact) with the vertical jump, only a single contact event was determined for the jump-stop - identified by the rising edge of the tibial origin longitudinal (+Y) acceleration (see Figure 43).

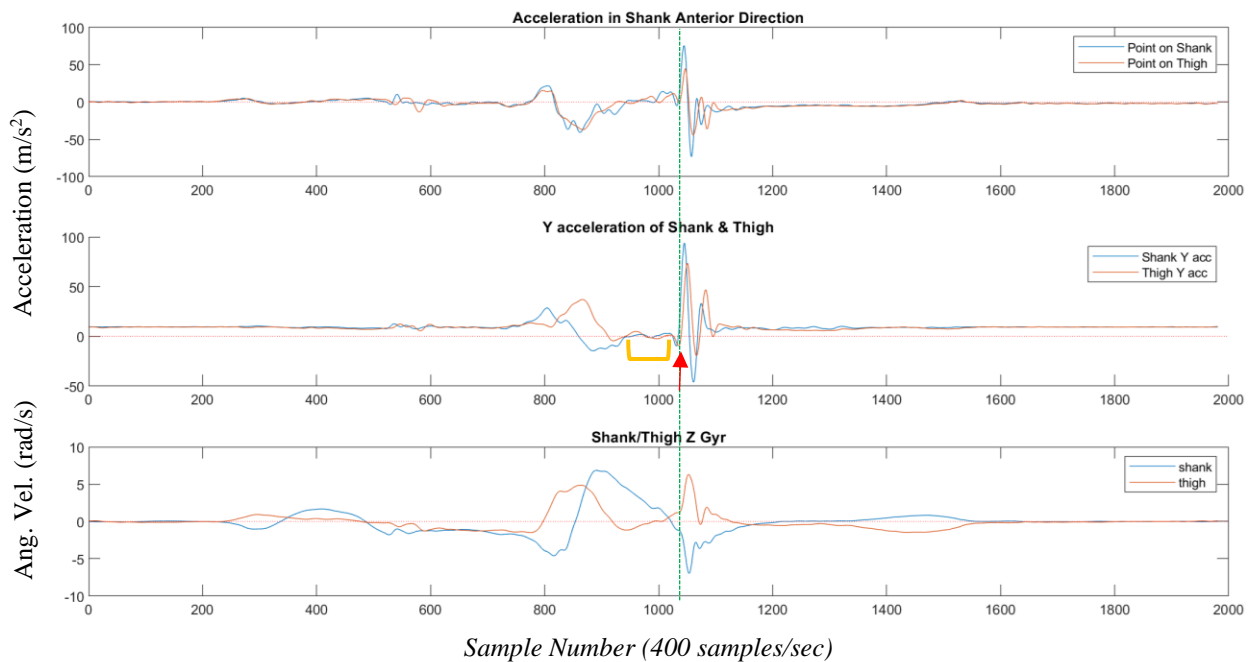


Figure 43 Example of a jump-stop free-fall period (yellow bracket) which occurs just prior to initial contact (red arrow) which is defined by the sharp rising edge of the tibial origin longitudinal (Y) signal.

Run and Cut: The initial ground contact for the run and cut activity was defined as the ground contact when the plant leg, with sensor attached, contacts the ground. This event was identified by using tibial origin acceleration in the posterior (-X) and longitudinal (+Y) directions. The most negative tibial origin acceleration valley occurred near initial ground contact. The rising edge of the peak longitudinal tibial origin acceleration nearest to the posterior acceleration event was used to define the time of initial ground contact (see Figure 44).

Sprint to Backpedal: The initial ground contact for the sprint to backpedal activity was defined as the ground contact of the leg with sensors attached that produced the most negative tibial origin acceleration. The rapid deceleration was assumed to be the most taxing part of the movement on the ACL and therefore was the portion of data analyzed and RATA calculated. The rising edge of the peak longitudinal (+Y) tibial origin acceleration nearest to the posterior acceleration event was used to define the time of initial ground contact (see Figure 45).

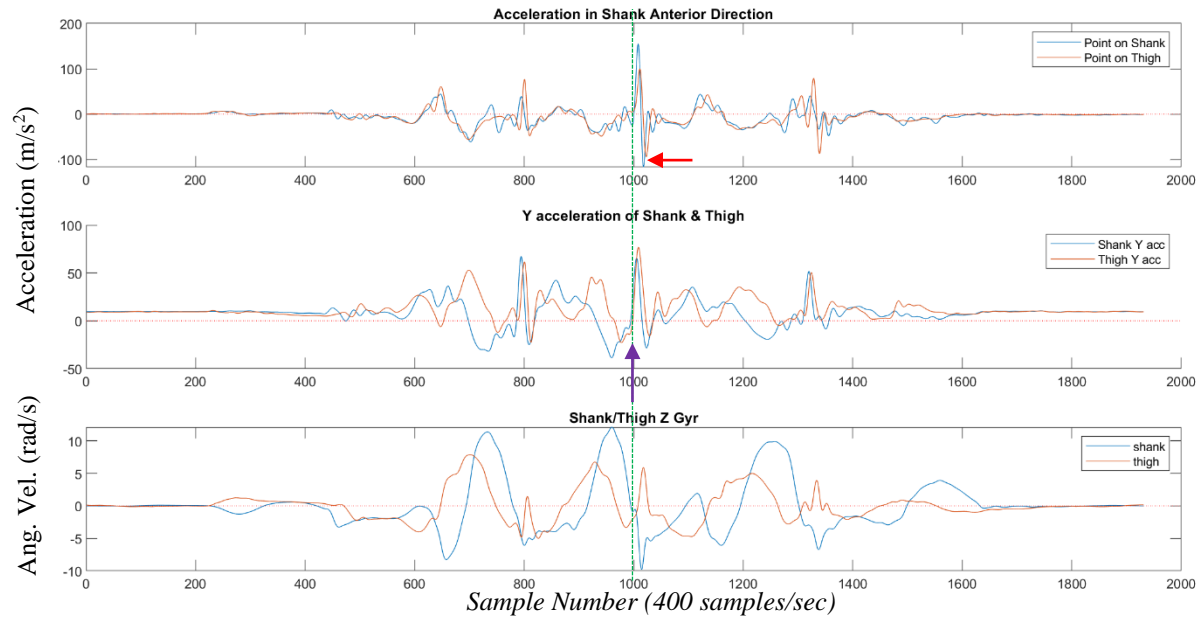


Figure 44 Example of acceleration data from a run and cut activity trial. The most negative tibial origin acceleration in the posterior direction (red arrow) helps identify which longitudinal acceleration peak to use to define initial contact. The longitudinal acceleration peak nearest in time to the posterior acceleration valley is identified and the rising edge of this peak (purple arrow) defines initial contact.

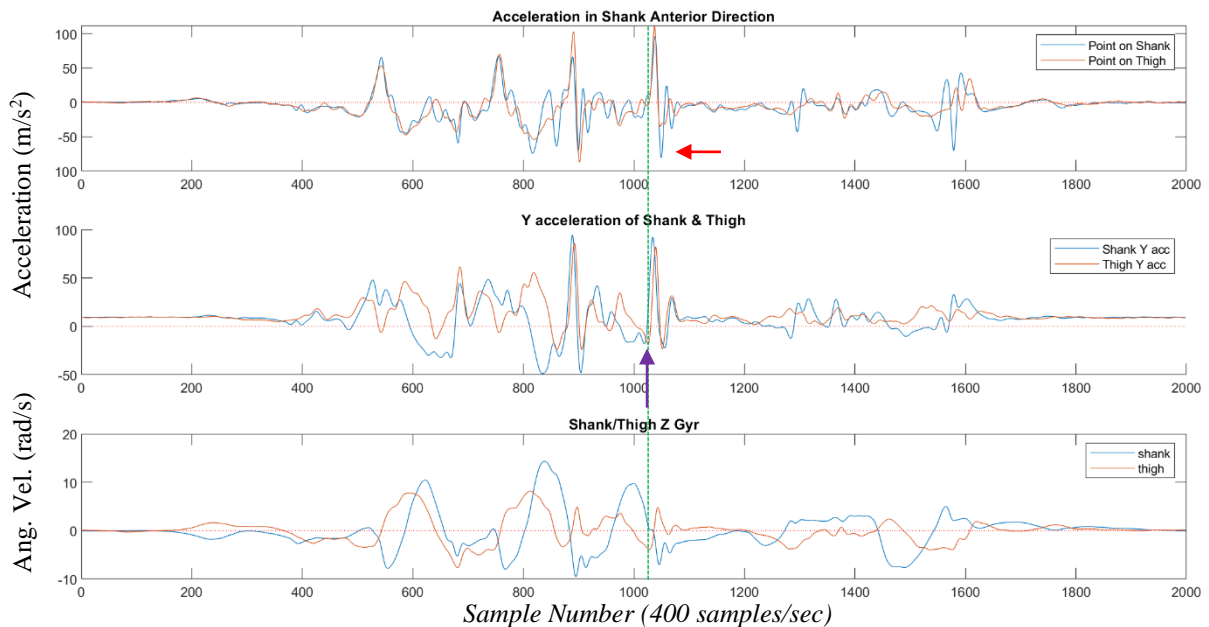


Figure 45 Example of acceleration data from a sprint to backpedal activity trial. The most negative tibial origin acceleration in the posterior direction (red arrow) helps identify which longitudinal acceleration peak to use to define initial contact. The longitudinal acceleration peak nearest in time to the posterior acceleration valley is identified and the rising edge of this peak (purple arrow) defines initial contact.