UNIVERSITY OF ALBERTA

FACULTY OF ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

Progress report

Temporal-spatial parameters in in-line roller skating using wearable sensors

Authors:

AMINREZA KHANDAN - ARMIN NOROUZI YENGEJE

(Contribution: 50% each)

June 22, 2020

FACULTY OF
ENGINEERING
UNIVERSITY OF ALBERTA

Contents

1 Introduction

1.1 Background

In the United States, there were over 11 million roller skating participants in 2017 [1]. To the best of our knowledge, the majority of the studies published on roller skating are the epidemiology of injuries in these participants [2]. However, the kinematics and kinetics leading to these injuries were generally not studied. The first step to obtaining knowledge about roller skating biomechanics is to measure the participants' Temporal Spatial Parameters (TSPs) during the movements.

In 2012, a motion capture system was used to calculate the speed and metabolic rate of seven elite roller ski skating participants [3]. Also, Oxygen pro apparatus is used to measure the participants' work rate and metabolic rate. Based on the results, in a definite work rates, gross efficiency on steeper slopes is higher than the lower ones.

Roller skating is a pretty fast activity and needs ample space to perform. Therefore, common motion capture labs are not quite capable of measuring the biomechanics of "real" over-ground skating. Also, the concern of blocking markers behind the body during skating can dissuade the researchers from using the motion capture system to measure temporal, spatial parameters. Furthermore, the realism and accuracy of using a treadmill are disputable [4]. In those applications, wearable sensors can be an alternative to measure TSPs during skating without the fear of the speed of activity and the concern of space constraints and missing markers.

Myklebust et al. (2015) investigated a method to see how accurately can a single IMU sensor estimate the displacement of the sacrum (S1) of a person during ski skating [5]. Six world-class roller-ski-skating participants skated on a treadmill using two techniques. They noted that to secure RMS error of less than 8mm with respect to the marker trajectories, a combination of accelerometer and gyroscope is necessary. They also added using acceleration alone give the vertical displacement sufficiently accurate.

A novel method to detect foot-ground contact time during running is introduced in [6].

Measuring contact time during indoor activity using force-plates is commonplace in the lab environment, but during outdoor activity can make a lot of trouble. For this study, the tri-axial accelerometer is mounted on the tibiae of right legs of six subjects running from a stationary start to steady-state running. The contact time calculated using accelerometer was compared with a force platform. The results show a strong correlation between these two data-sets and the body mounted accelerometer can estimate the contact time correctly.

An on ice-measurement to investigate the forward skating technique differences between players with different skills is tested [7]. In these experiments, a 3D accelerometer mounted on the right skates to detect skate contact and then to calculate temporal-spatial parameters like speed is used. Moreover, by identifying strides, they differentiate between the accelerating phase and the steady-state phase to compare biomechanical parameters between these phases. They could implement an on-ice measurement approach using accelerometers to detect ice-skating events and evaluate biomechanical parameters during skating.

Most of the papers published on the topic of roller skating are about skating injuries epidemiology (for instance [2, 8]). However, as far as we know, no significant research investigates the etiology of these injuries, and the kinematics of roller skating has not been studied yet. None of the reviewed studies uses IMU sensors on roller-skating in field measurements. However, with consideration of the similarities between ice skating and roller skating, this literature review asserts this point that measuring temporal-spatial parameters with a combination of accelerometer and gyroscope is possible.

1.2 Objective

Application of wearable sensors are not confined to the laboratory environment and can be used in field measurements. These sensors in gait analysis are widely accepted system [9], but its application in other activities should be further investigated. Inertial Measurement Units (IMUs) are appropriate devices for long-term measurements since they have a large capacity for the captured data as well as their long battery life. Furthermore, these sensors

are small and low-weight and not restricting the degree of freedom of skating participants.

The objective of this study is to investigate temporal, spatial parameters of roller skating during an over-ground movement. The initial step in understanding roller skating biomechanics is to find these TSPs. They provide us the essential information of the skating performance and can be used to define indices correlating with onset of fatigue, a typical precursor to dropping in skating participant's performance and injuries. These parameters can also be applicable to differentiate between amateur and professional participants [7]. Getting familiar with this discerning parameters can help the coaches to prepare more efficient training sessions for juvenile trainees.

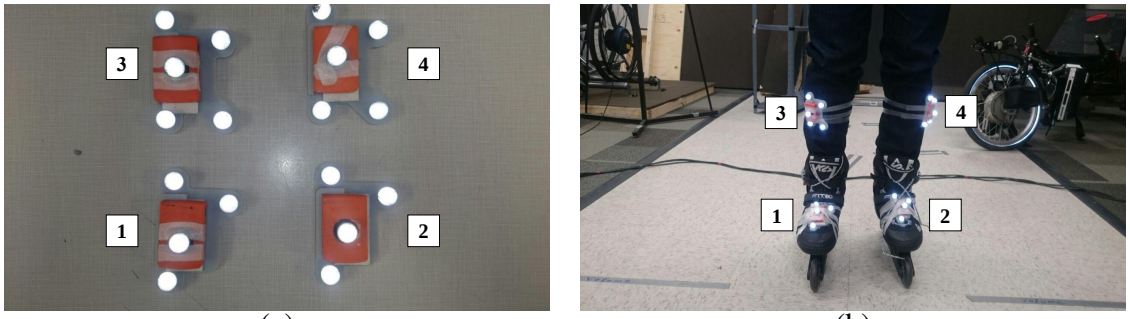
2 Data Acquisition

2.1 Experiment Design

Four IMU sensors from Xsens Technologies B.V (Enschede, The Netherlands) by double-sided tapes were stuck tightly on plates with four to five retro-reflective markers. The plates placing on shanks have five markers, and the ones on foot have four retro-reflective markers (Figure 1. a). These markers can be used either as a gold standard for the position of the segment or to calibrate the IMU reference frame system concerning the global coordinate system (GCS, Lab Coordinate system).

The acceleration of the segments was acquired by mounting IMU sensors on the shank and skate of each of the skating participant's leg (Figure 1. b). Besides, trajectories of retro-reflective markers were obtained using 8-cameras motion capture system (VICON, Oxford Metrics Group, Oxford, U.K., 100 Hz). By using these eight cameras, the most significant possible volume is captured in the lab environment. Camera position configuration was determined such that not only they captured as large amount as possible, but also they placed where minimally interfere with the participant's skating. We asked the participant to skate alongside the length of the motion capture lab. These experiments were repeated five times

in a session. After the end of the experiments, the marker was labeled, and the gaps in which markers missed during the tests were filled using different interpolation algorithms in Vicon Nexus (VICON, Oxford Metrics Group, Oxford, UK). Derived marker trajectories will be used to calculate temporal-spatial parameters.



(a) (b)

Figure 1: a. Sensors on the plates with markers, b. Experiment Setup

2.2 Signal Processing Theory on Data Acquisition

Discrete Sampling Frequency (DSF) of the output of the acceleration using Fourier Fast Transform (FFT) was obtained. Using this technique, the highest significant frequency content of the IMU acceleration signals was $\approx 10Hz$. Also, there is an anti-aliasing filter in our Xsens IMU sensors. The sampling frequency of the motion capture system was also 100 Hz. This frequency is also able to sample the marker trajectories. According to [10] the highest frequency content of the markers mounted on human bodies is 2 Hz. Thus, based on Nyquist (Sampling) theorem [1], 100 Hz is an appropriate frequency to measure the essential content of the acceleration.

The quantization resolution of the IMU sensors is 16-bits. The IMU uses an internal 16-bit A/D converter to converts measured analog acceleration signals to digital ones. The data obtained from IMU is comprised of high-frequency variations due to the agility of skating and motion artifacts due to body movements. The quantization step during A/D conversion

[1] Sampling frequency must be higher than highest frequency present in the signal itself [11]

is calculated as:

$$Q = \frac{R}{2^N} \tag{1}$$

where R and N are full-scale ranges of analog inputs and number of bits used in A/D converter, respectively. So, the quantization step for accelerometer and gyroscope are calculated as:

$$Q_{acc} = \frac{2 \times 160}{2^{16}} = 0.0049\ m/s \tag{2}$$

$$Q_{gyr} = \frac{2 \times 34.9}{2^{16}} = 0.0011\ rad/s \tag{3}$$

Additionally, the dynamic range of the converter is computed as

$$Q_{dr} = 20\ log(2^N) = 20\ log(2^{16}) = 221.81\ dB \tag{4}$$

3 Raw Digital Signals

3.1 Free Acceleration

To obtain the displacement from the acceleration, we need to eliminate gravitational acceleration (g [m/s^2]) and then numerically take integral the data twice. For this purpose, free acceleration (By eliminating g) expressed on the global coordinate system is needed. The Xsens sensors provided us this free acceleration. Figure 2 illustrated free acceleration of x, y, and z-direction of sensor 1 and 2. Besides, all the MATLAB codes are available in Appendix G.

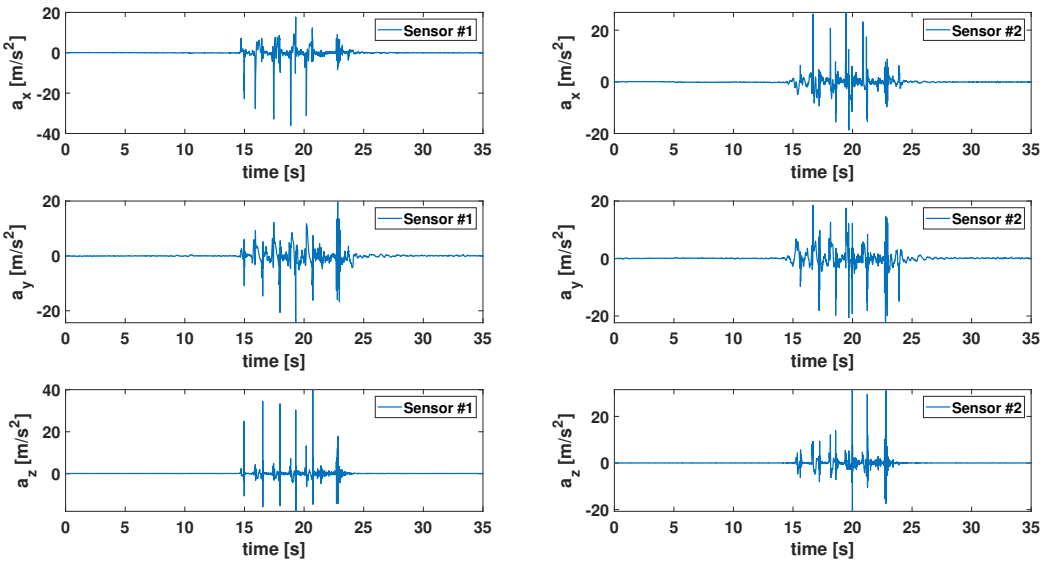


Figure 2: Raw data from sensors 1 and 2

3.2 Frequency Domain Representation

As our raw data are a non-periodic signal, the best representation in the frequency domain is Fourier Fast Transform. Figure 3 shows the frequency content of all sensors in x, y, and z-direction. $P_x(f)$, $P_y(f)$, and $P_z(f)$ are the amplitude of the signal in the frequency domain for x, y, and z-direction, respectively. As discussed earlier in the previous section, based on the Nyquist Sampling Theorem a continuous-time signal can be perfectly reconstructed from its samples if it is sampled over twice as fast as it's highest frequency component. As shown in Figure 3 informative features of the signal is below 10 Hertz. So, the 100 Hertz sampling frequency is high enough to avoid aliasing.

3.3 Markers Trajectories of the plates

Marker positions of each sensor were captured using a motion capture system. Right foot, right shank, left foot, and left shank markers trajectories are illustrated in Appendix D. As explained in the previous section, for each of the shank and foot sensors we have five and four markers attached to the plates. Therefore, these marker trajectories in each plate were

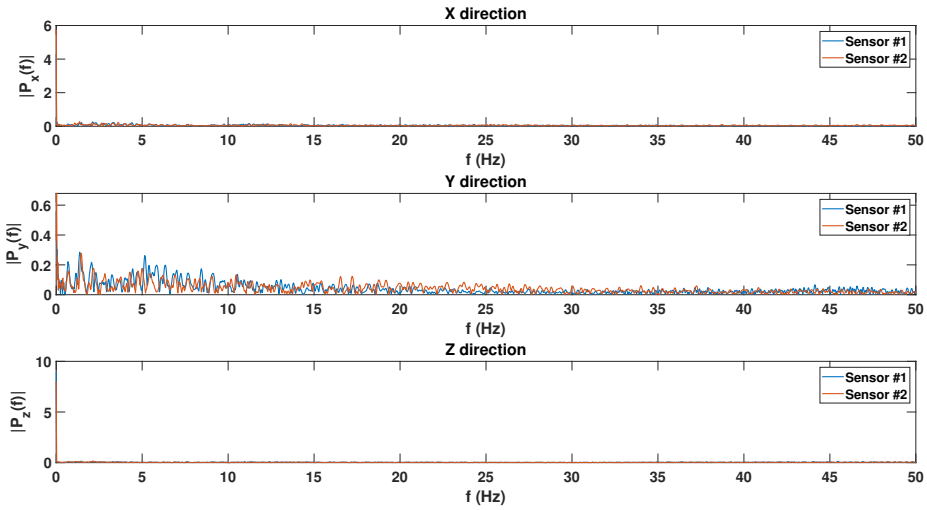


Figure 3: Fast Fourier transform of signals

almost parallel with each other during the skating.

3.4 Potential Sources of Noises

One potential source of noise in this experiment is high-frequency power lines and other electrical devices in the lab environment. Moreover, analog to digital (A/D) conversion and saving into binary format can also cause fluctuations. Furthermore, vibration, soft tissue artifacts, and misalignment errors are other conventional sources of error that may appear at these types of experiments.

4 Method

We used IMU sensors data to detect critical events and thereby temporal-spatial parameters of the skating participant. In the first step, noises due to unwilling body movements artifacts, the power lines and other sources of noise were removed. The IMU sensors displacements will be obtained by taking double integral of free acceleration (acceleration by eliminating **g**) in the global coordinate system. Important events such as skate-strike and skate-off are detected. Then, by the times and locations of these events, skating participant's temporal-

spatial parameters including contact time, speed, step and stride time and length were calculated. Hereafter in this section, these stages were described in details.

4.1 Filter Design

4.1.1 FIR filters

FIR filter is designed based on the pass-band (PB) edge, stop-band (SB) edge frequency, and stop-band attenuation. Three windows Hanning, Hamming, and Kaiser were chosen with the same PB edge frequency, SB edge frequency and SB attenuation to confine the infinite filter. The filters will not cause phase distortion since odd numbers of coefficients chose for the filters. A MATLAB code was used to design the filters (Appendix G). FIR Filter shape, impulse response, step response can be found in Appendix F. Figures 4 and 5 illustrate raw signal versus filtered signal using FIR filters with Hanning, Hamming, and Blackman windows (Bottom plots are the zoomed version of top plots). It can be seen that

Table 1: FIR Low pass filter specification

Specification $\backslash FIRwindowstype$	Hanning	Hamming	Kaiser
SB attenuation (DB)	44	55	75
SB edge frequency	$20Hz$	$20Hz$	$20Hz$
PB edge Frequency	$10Hz$	$10Hz$	$10Hz$
Number of Coefficients	33	35	61
Monotonic in PC	No	No	No
Monotonic in SB	No	No	No
Phase Distortion	none	none	none
Stability	Stable	Stable	Stable

low-pass FIR filtering has successfully attenuated high-frequency noises. In addition, the overall pattern of the signal was almost the same, and the meaningful peaks were kept intact. These filters did not change the location of the peaks and hardly affected the magnitude of these peaks. Also, implementing different types of windows led to almost same filtered free acceleration signals. Based on the results, all the window performance are practically the same, so the best choice is Hanning window, the simplest one.

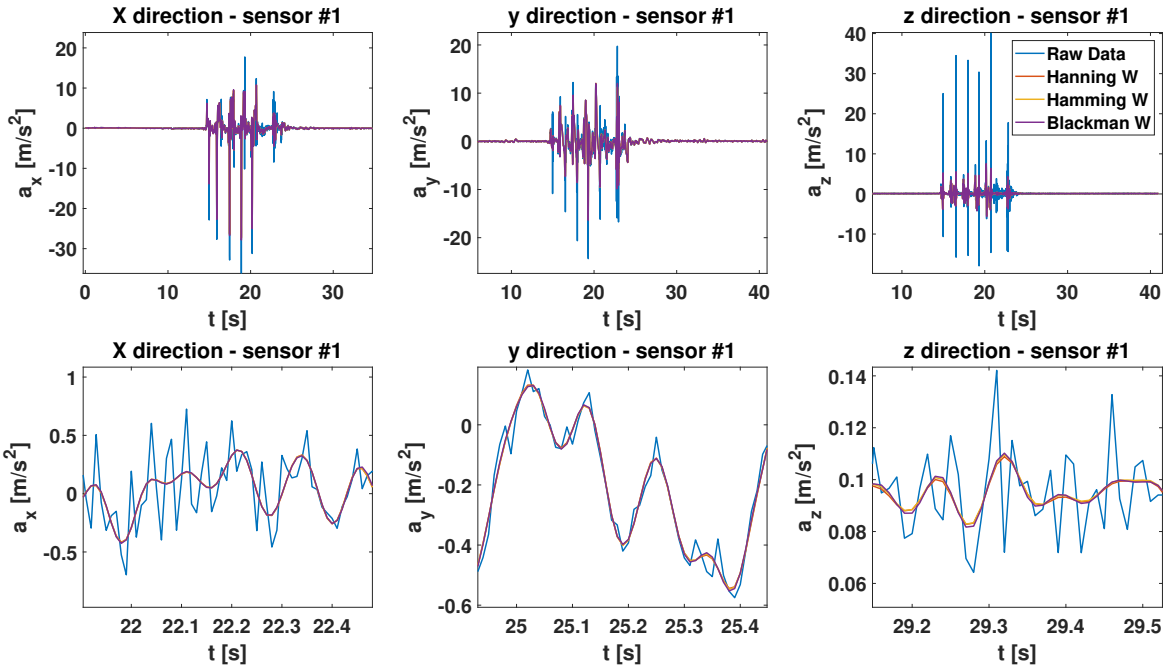


Figure 4: Raw signals vs Filtered signals of sensor number 1 using different FIR filters

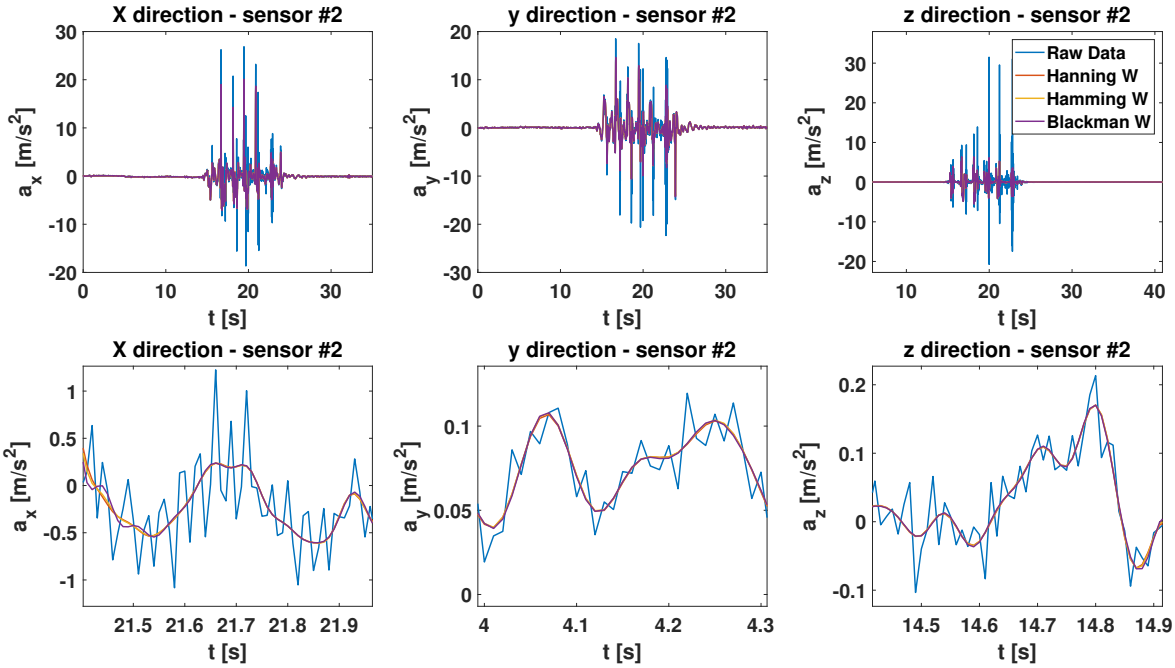


Figure 5: Raw signals vs Filtered signals of sensor number 2 using different FIR filters

4.1.2 IIR filters

For implementing IIR filters, two types of filters were selected for this study: Butterworth (BW) and Chebyshev type I (CS). BW filter Using MATLAB built-in functions was designed based on its order and normalized cut-off frequency. PB edge frequency in BW filters was predetermined to be $-3dB$. Moreover, the CS filter type I was designed according to its order, PB edge frequency, and PB ripple. IIR Filter shape, impulse response, step response can be found in Appendix F. Figures 6 and 7 illustrate raw signal versus filtered signal using Chebyshev type I and Butterworth filters (Bottom plots are the zoomed version of top plots). CS type I followed the pattern of the raw velocity signal more than BW filter. However, in terms of skating participant's displacements and TSPs, these two filters barely affected the results.

In conclusion, Since the IIR filters designed for this study are stable and needs less coefficient, our final decision for the type of the filter is IIR filters. Moreover, the smoothness is a critical decision factor in the field of Biomechanics. Taking this into account, we select the BW filter for this study. The filter shape, impulse, and step response is presented in Appendix F.

Table 2: IIR Low pass filter specification

Specification - IIR filter type	Chebyshev Type I	Butterworth
Order	4	4
Cut-off Frequency	–	10 Hz
PB Edge frequency	–	–
PB Ripple (DB)	0.001	–
Number of Coefficients	8	8
Monotonic in PC	No	Yes
Monotonic in SB	Yes	Yes
Phase Distortion	Yes	Yes
Stability	Checked	Checked

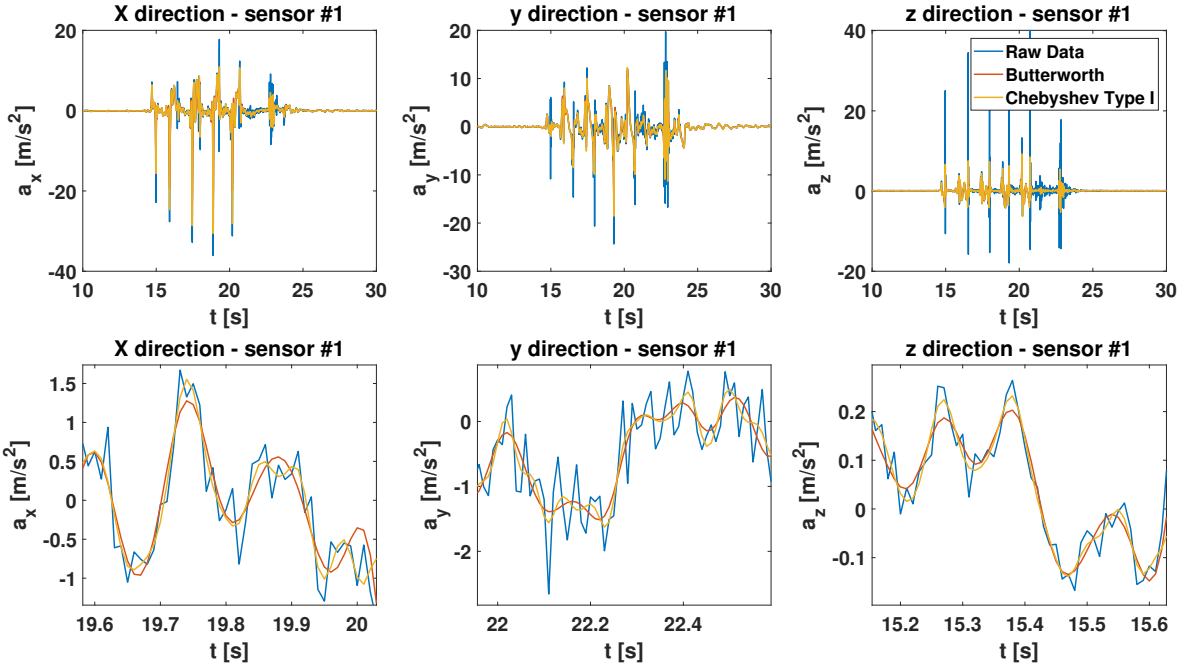


Figure 6: Raw signals vs Filtered signals of sensor number 1 using different IIR filters

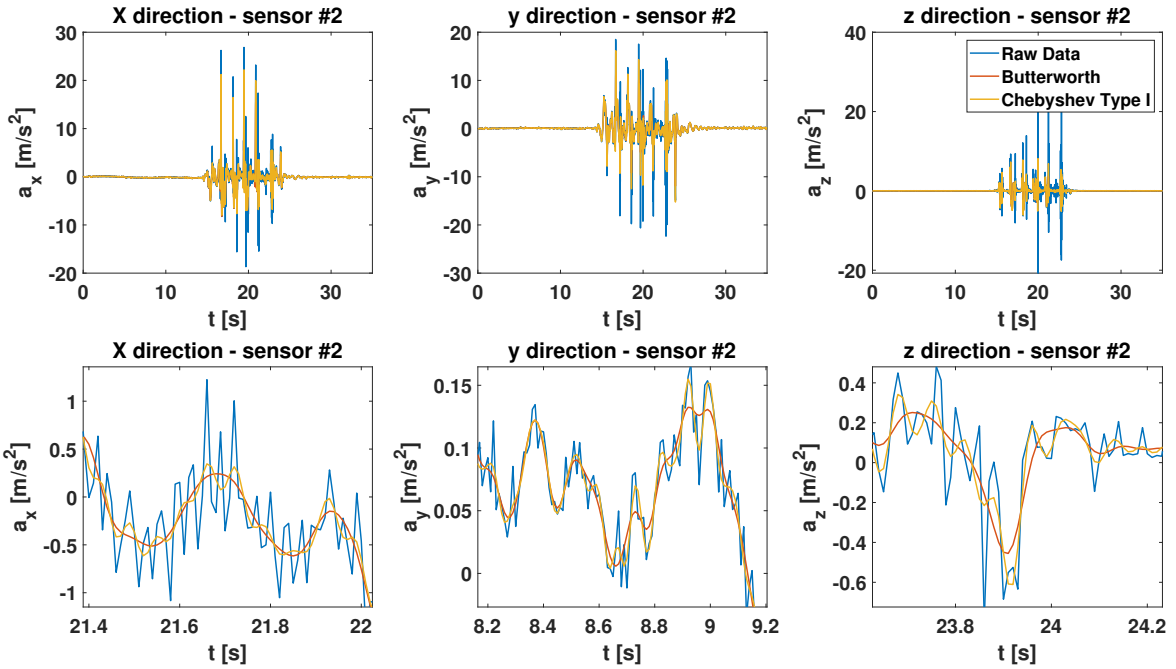


Figure 7: Raw signals vs Filtered signals of sensor number 1 using different IIR filters

4.2 Event Detection

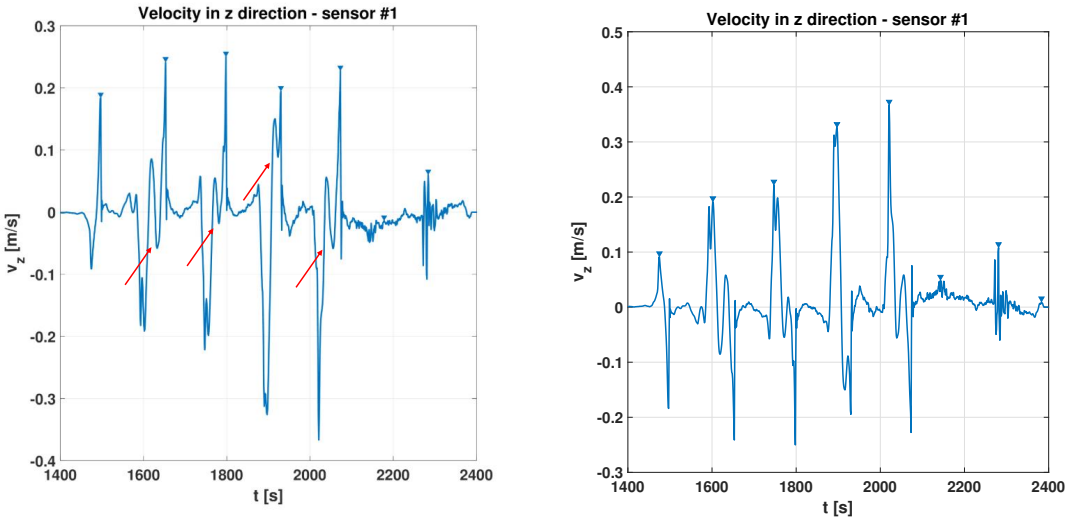
The initial step to calculate TSPs is detecting important events during skating. These events are skate strikes and skate off for both skates.

- **Skate Strike Detection**

Skate strike is the moment when the skate hits the ground. As a consequence, the vertical velocity of the skate will change direction at once. These events were automatically detected using the speed of right and left skates derived by taking integral of the free acceleration over time. This time is chosen to be the peak of a negative velocity of the skate(Figure 8-b).

- **Skate-off**

Skate-off is the time that skate goes off the ground. In these moments, the vertical velocity experiences a significant jump to its maximum values. Therefore, the minimums before the highest maximums of the skating participants were chosen as Skate-offs (Figure 8-a).



(a) (b)

Figure 8: Velocity of sensor 1 in upward (a), and downward (b) direction

4.3 Temporal Spatial Parameters

TSP during skating are the ones provide us general information of the skating participant's temporal and spatial conditions in his/her repetitive movements. In this study, we investigated

the following parameters:

Step length and step time:

Step is defined as the consecutive skate strikes of right and left skates concerning each other. The relative time of these events are called step time, and the distance between these two events are step length.

Stride length and time:

Stride is defined as the consecutive skate strikes of right or left skates. The stride time and stride length is defined similarly.

Skating participant's speed:

The stride length over the stride time of one period of skating is the skating participant's speed. These speeds can be the same in both legs or not.

5 Results of Temporal Spatial Parameters

The displacement signals were calculated (see Appendix C for correction strategies and see Appendix D for the velocities and displacements graphs). In Table 3 the TSPs using the sensors' velocities and displacements signals were presented. This table indicates that contact time and step time of the right (dominant) leg is more than Left Leg. Stride time of both of the legs was almost the same. In addition, standard deviation of temporal parameters are less than spatial parameters. Therefore, these parameters are almost steady in the trials.

Table 3: Temporal Spatial parameters of the roller-skating participant

Parameters	Right	Left
Step time (s)	0.94 ± 0.05	0.5 ± 0.12
Stride time (s)	1.4 ± 0.06	1.44 ± 0.15
Step length (m)	0.50 ± 0.38	1.15 ± 0.20
Stride length (m)	2.01 ± 0.37	1.65 ± 0.58
Speed (m/s)	1.44 ± 0.27	1.12 ± 0.32
Contact time (s)	1.00 ± 0.03	0.80 ± 0.03

6 Discussion

Implementing different filters removed the noises of the free acceleration signals. FIR filters, regardless of the types of the windows, removed the noises due to motion artifacts and some of the integral drifts. The filters did not affect the position of a global extremum in time-line and also on their magnitudes. The same is persisting for IIR filters. Our final decision is a Butterworth filter which widely applicable in Biomechanics [11].

Comparing the results with the parameters calculated by motion capture shows that the event detected using the vertical velocity were accurate (see Appendix C). Stride length are almost the same for right leg (2.01 ± 0.37 vs 2.05 ± 0.39) and 21 percent less in left leg (2.01 ± 0.37 vs 2.05 ± 0.39). However, the drifts due to double integral over time influenced data and cause the difference between these two data-sets. In addition to our correction strategies, more external resources are needed to correct the velocity and reduce the drifts due to taking integral.

The system of two IMU sensors placed on the skates of the skating participants was capable of providing us the temporal-spatial parameters of the skating participant. Hence, the Sensors on the shank is not essentially needed in this system. As Purcell noted the contact time can be measured using one accelerometer [6]. However, in contrast with Myklebust's findings [5] the spatial parameters are not accurate with respect to MoCap data. It seems that a fusion with gyroscope data is required to find the spatial parameters with higher accuracy.

The first limitation of this study is using external resources to modify the sensors' velocities and displacements. Using the output of Gyroscopes or Magnetometers and fusing this free acceleration with these sensors may assist in getting a more accurate position for future studies.

The second limitation can be the presumption of walking on a straight line toward the North. This assumption is valid for our experimental setup and walkway. The Xsens sensors provide us the free acceleration on North-East-Up (NEU) reference frame. Therefore, if the skating participant deviates from this direction, the location of the sensors in both x- and

y-directions should be considered.

We changed the configuration of the cameras to obtain the highest captured volume. Nevertheless, this volume captured by motion capture devices because of limitations in numbers of cameras were almost half on the skating participant's skating volume. As a result, we could validate half of the strides in each trial.

7 Conclusions and Future Works

The objective of this study was to investigate temporal, spatial parameters of roller skating during an over-ground movement. The system of two IMU sensors placed on the skates of the skating participants was capable of providing us the essential information of the skating. Using the free acceleration of the IMU sensors and taking the integral of these signals can give us the velocity of the skates which can be used to detect important skating events such as skate-strike and skate-off during the movement. Applying modification and then taking integral of the modified velocity signal, the displacement of the skates can be calculated. Although the accuracy of the system can be higher, this system can be a first step in obtaining more information about skating kinematics.

For future studies, Sensor Fusion with 3D gyroscopes or magnetometers may suggest an efficient way to reduce the requirement of external resources. Moreover, getting data from more participants will help to reach a more reliable conclusion. Finally, similar to the procedure of gait analysis using wearable sensors, the modification process can be performed in each step. In gait analysis using wearable sensors, the anterior-posterior velocity of the foot in stance is set to be zero. The big challenge of applications of IMU sensors in skating is to find another correction factor for the skating participants.

References

[1] O. Foundation, "Number of participants in roller skating in the united states from 2006 to 2017 (in millions)," *Online survey*, 2017.

[2] S. C. Calle and R. G. Eaton, "Wheels-in-line roller skating injuries.," *The Journal of trauma*, vol. 35, no. 6, pp. 946–951, 1993.

[3] Ø. Sandbakk, G. Ettema, and H.-C. Holmberg, "The influence of incline and speed on work rate, gross efficiency and kinematics of roller ski skating," *European journal of applied physiology*, vol. 112, no. 8, pp. 2829–2838, 2012.

[4] D. Lafontaine, "Three-dimensional kinematics of the knee and ankle joints for three consecutive push-offs during ice hockey skating starts," *Sports Biomechanics*, vol. 6, no. 3, pp. 391–406, 2007.

[5] H. Myklebust, Ø. Gløersen, and J. Hallén, "Validity of ski skating center-of-mass displacement measured by a single inertial measurement unit," *Journal of applied biomechanics*, vol. 31, no. 6, pp. 492–498, 2015.

[6] B. Purcell, J. Channells, D. James, and R. Barrett, "Use of accelerometers for detecting foot-ground contact time during running," in *BioMEMS and Nanotechnology II*, vol. 6036, p. 603615, International Society for Optics and Photonics, 2006.

[7] E. Buckeridge, M. C. LeVangie, B. Stetter, S. R. Nigg, and B. M. Nigg, "An on-ice measurement approach to analyse the biomechanics of ice hockey skating," *PloS one*, vol. 10, no. 5, p. e0127324, 2015.

[8] R. A. Schieber, C. M. Branche-Dorsey, and G. W. Ryan, "Comparison of in-line skating injuries with rollerskating and skateboarding injuries," *JAMA*, vol. 271, no. 23, pp. 1856–1858, 1994.

[9] T. Seel, J. Raisch, and T. Schauer, "Imu-based joint angle measurement for gait analysis," *Sensors*, vol. 14, no. 4, pp. 6891–6909, 2014.

[10] J. Abdul, "The shannon sampling theorem—lts various extensions and applications: A," *Proceedings of the IEEE*, vol. 65, no. 11, p. 1565, 1977.

[11] D. A. Winter, *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009.

[12] "Mtw awinda, wireless motion tracker," *ed: xsense*, 2018.

Appendix A: Technical specification

Table A1: Important technical specification of Xsens MTw Awinda [12]

Internal sampling rate	1000 Hz
Battery life (continuous use)	6 *hours*
Dimension tracker	47 $\times$ 30 $\times$ 13 *mm*
Full-charge Operation Time	8 *hours* $(typical)$
Mass	$16g(0.56\ oz.)$
Temperature Range	$0\ ^{\circ}C - 50\ ^{\circ}C$

Table A2: Tracker components of Xsens MTw Awinda [12]

b_0	Angular velocity	Acceleration	Magnetic field
Dimensions	3 *axes*	3 *axes*	3 *axes*
Full scale	$\pm 2000\ deg/s$	$\pm 160\ m/s^2$	± 1.9 *Gauss*

Appendix B: Signal Correction Strategies

These correction steps were applied on the signals:

- Implementing filter: The filters were introduced in section 4.1 removed most of the unwilling signals and obtaining high-quality acceleration signals.
- Correcting the beginning and the end of the velocity graphs. In the day of the experiments, we asked the participant to stand still in the first and the last ten seconds of the trials. Therefore, the velocity of the IMU sensors should be zero at those periods. We made the velocity to be zero in these periods.
- Removing the linear trend of the velocity. After correcting the first and the last ten seconds of the velocity data, a discontinuity is seen in the data. Besides, a linear trend in the velocity data is visible that is modified by detrending the velocity.
- Taking integral over the enter time domain: In this step we took the integral over time and calculated the displacement of the skating participant.
- Normalization of the total movement to the length of the walkway: To reach better results, the entire range of the displacement is set to be the length of the laboratory (10 m).

Appendix C: TSP based on marker trajectories

Table C1: Temporal Spatial parameters of the roller-skating participant based on marker trajectories

Parameters	Right	Left
Step time (s)	0.76 ± 0.01	0.58 ± 0.09
Stride time (s)	1.34 ± 0.01	1.34 ± 0.11
Step length (m)	1.26 ± 0.132	0.76 ± 0.18
Stride length (m)	2.05 ± 0.39	2.10 ± 0.04
Speed (m/s)	1.52 ± 0.30	1.57 ± 0.10
Contact time (s)	0.93 ± 0.1	0.83 ± 0.01

Appendix D: Position and velocity plots

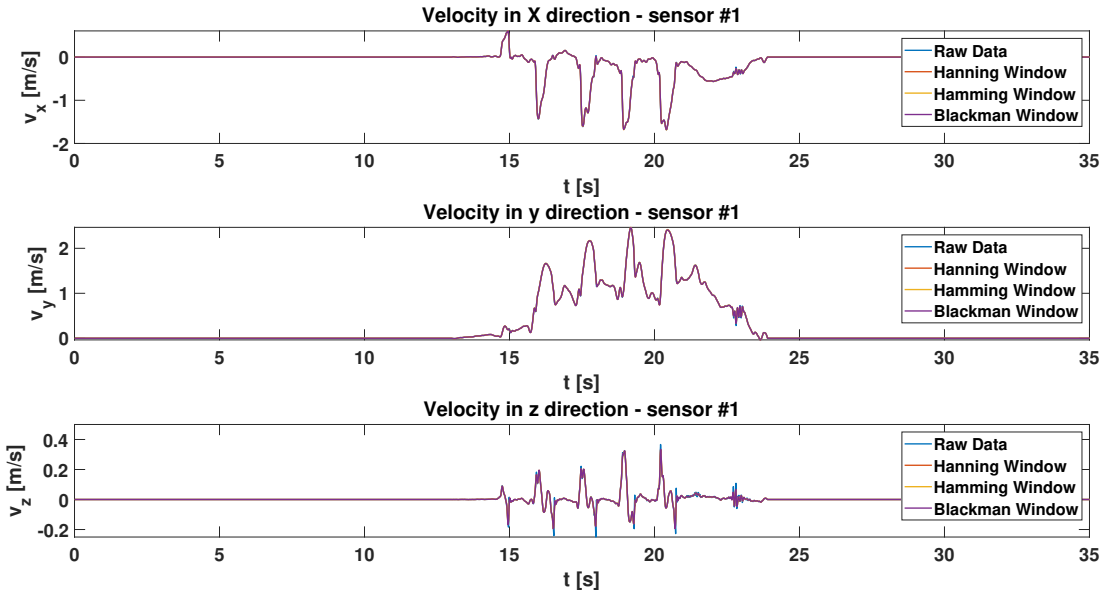


Figure D1: velocity FIR sensor number 1

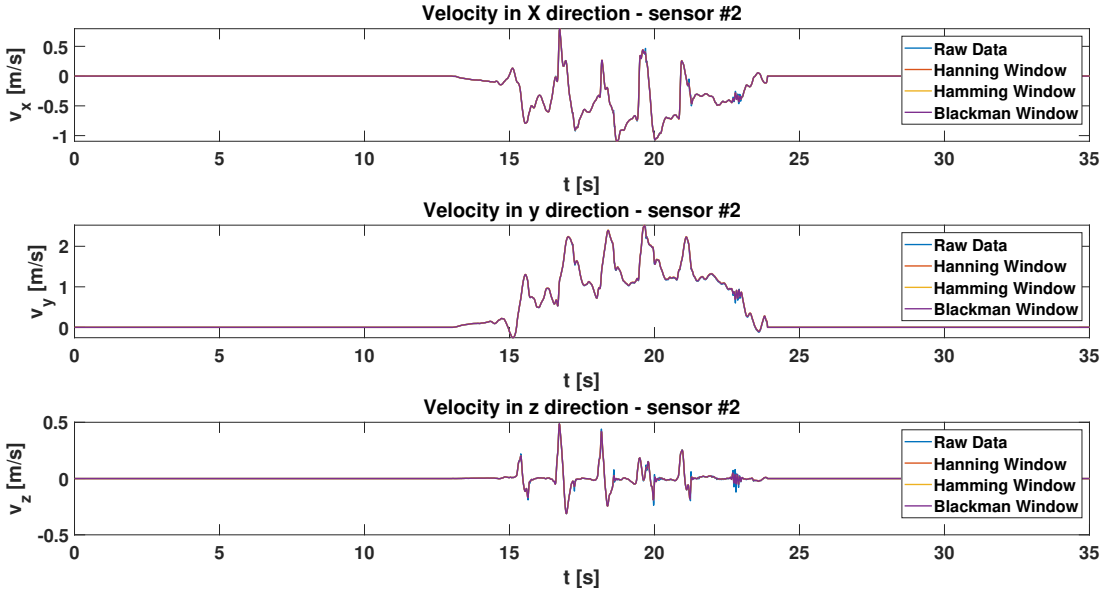


Figure D2: velocity FIR sensor number 2

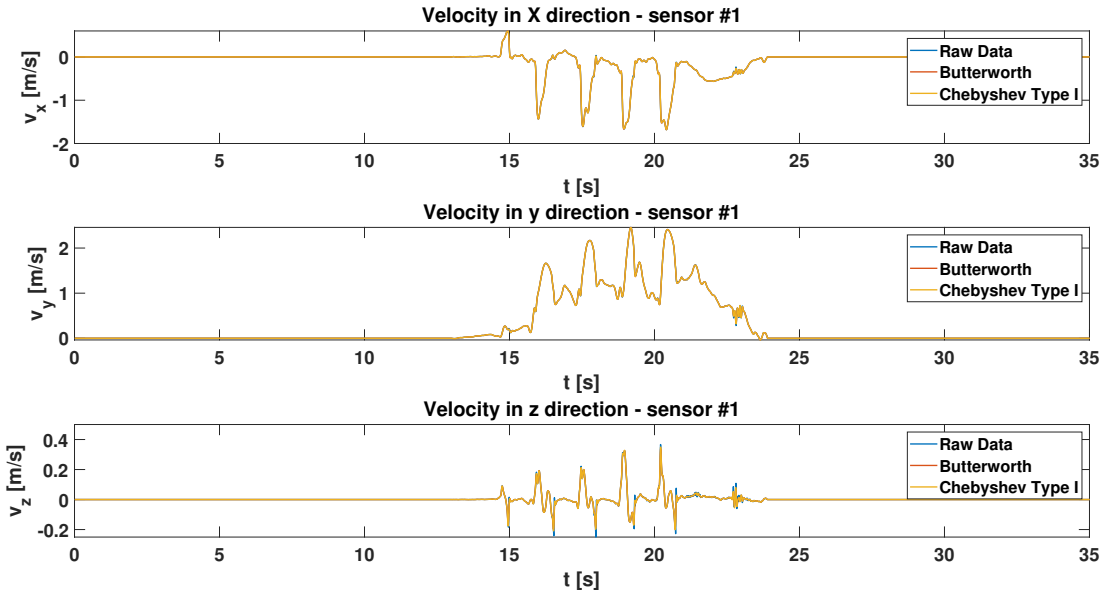


Figure D3: velocity IIR sensor number 1

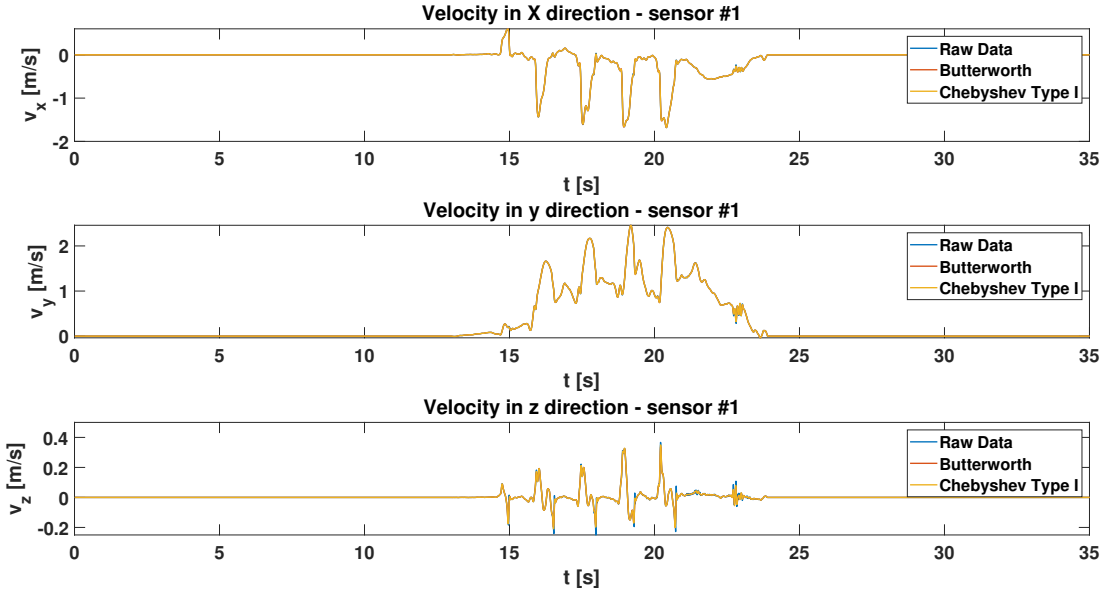


Figure D4: velocity IIR sensor number 2

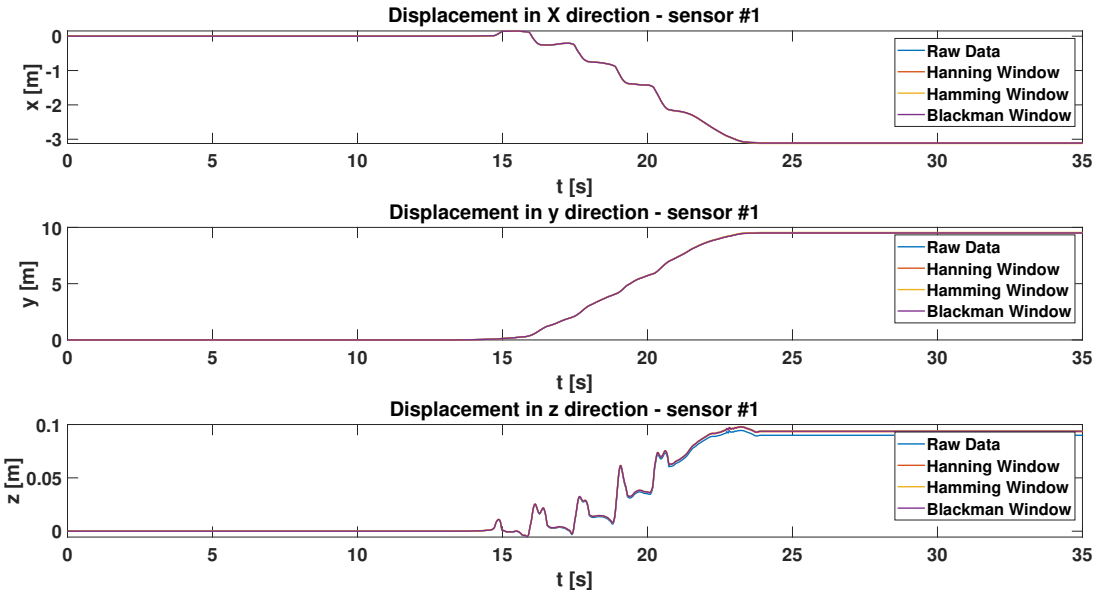


Figure D5: Position FIR sensor number 1

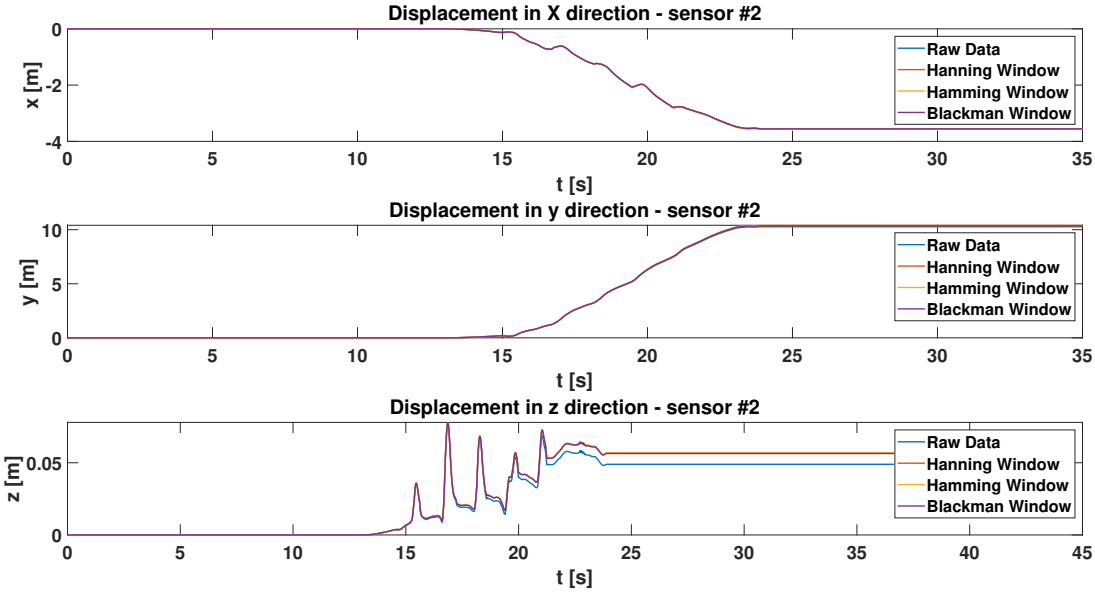


Figure D6: Position FIR sensor number 2

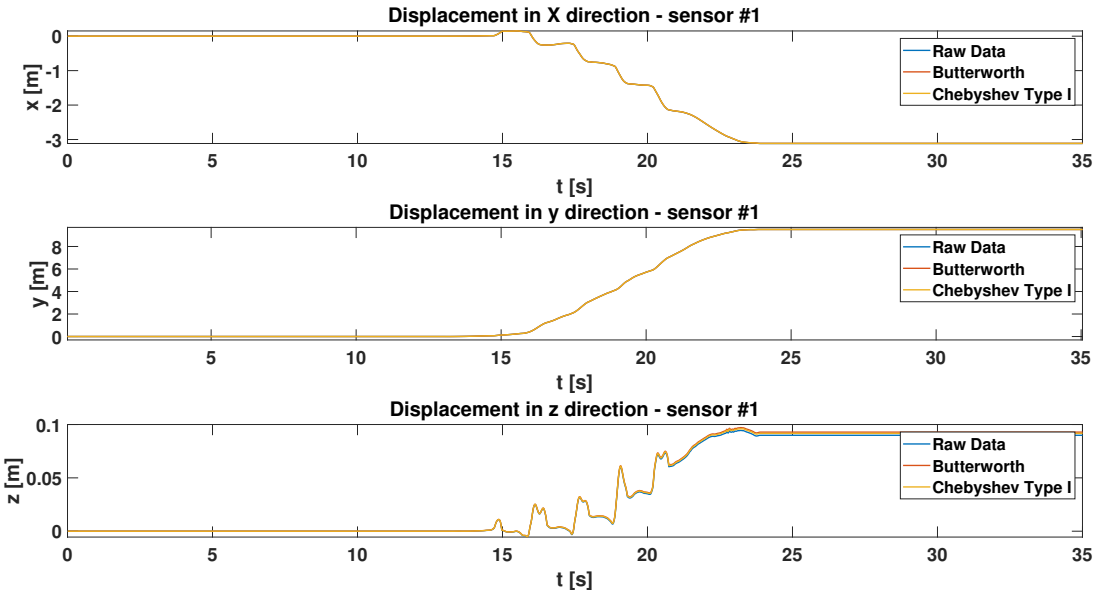


Figure D7: Position IIR sensor number 1

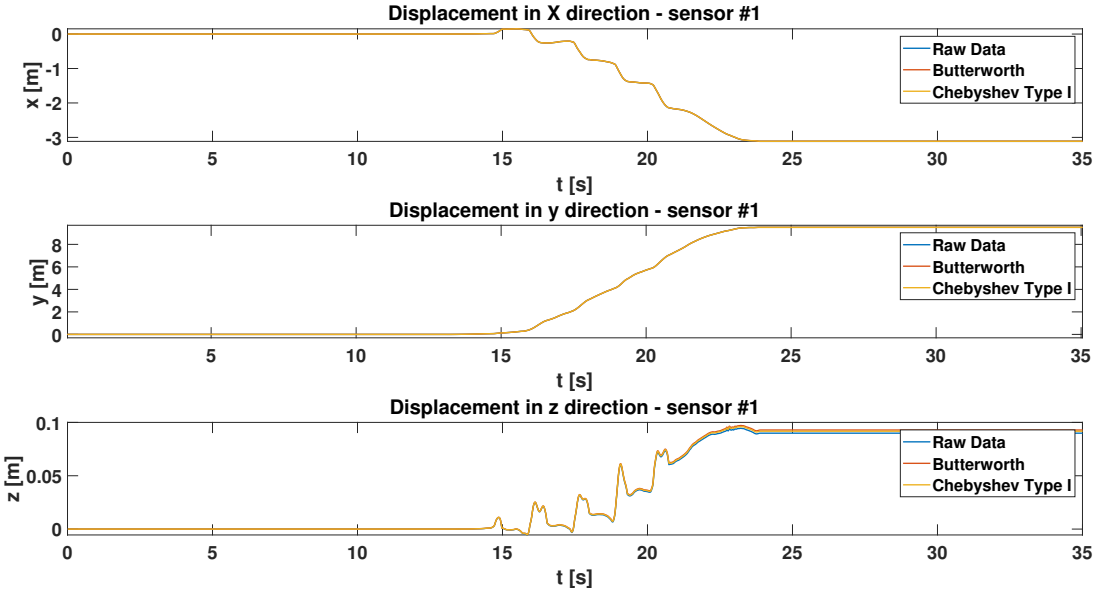


Figure D8: Position IIR sensor number 2

Appendix E: Marker Trajectories

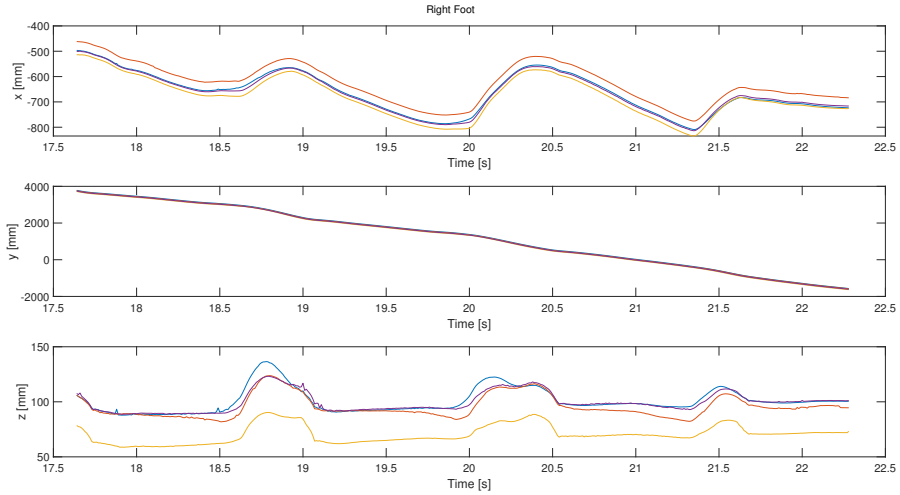


Figure E1: Location of Markers of sensor on the right foot (sensor 1)

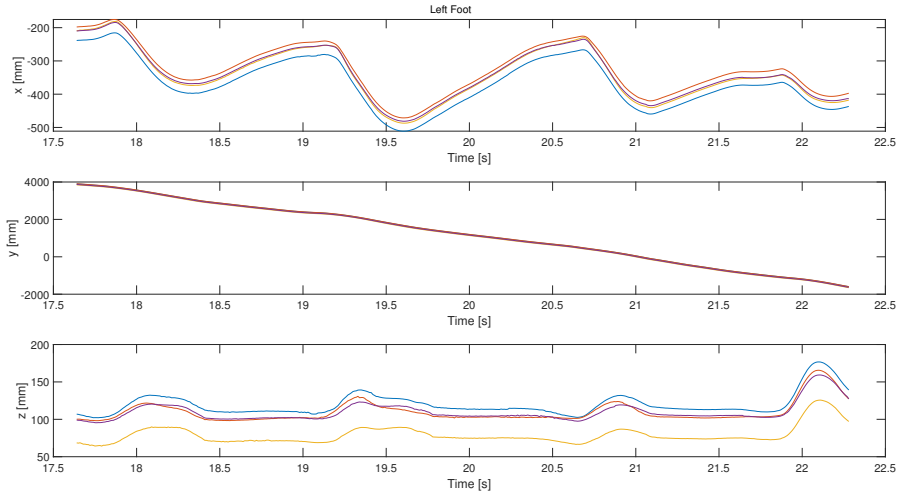


Figure E2: Location of Markers of sensor on the left foot (sensor 2)

Appendix F: Filters Configuration

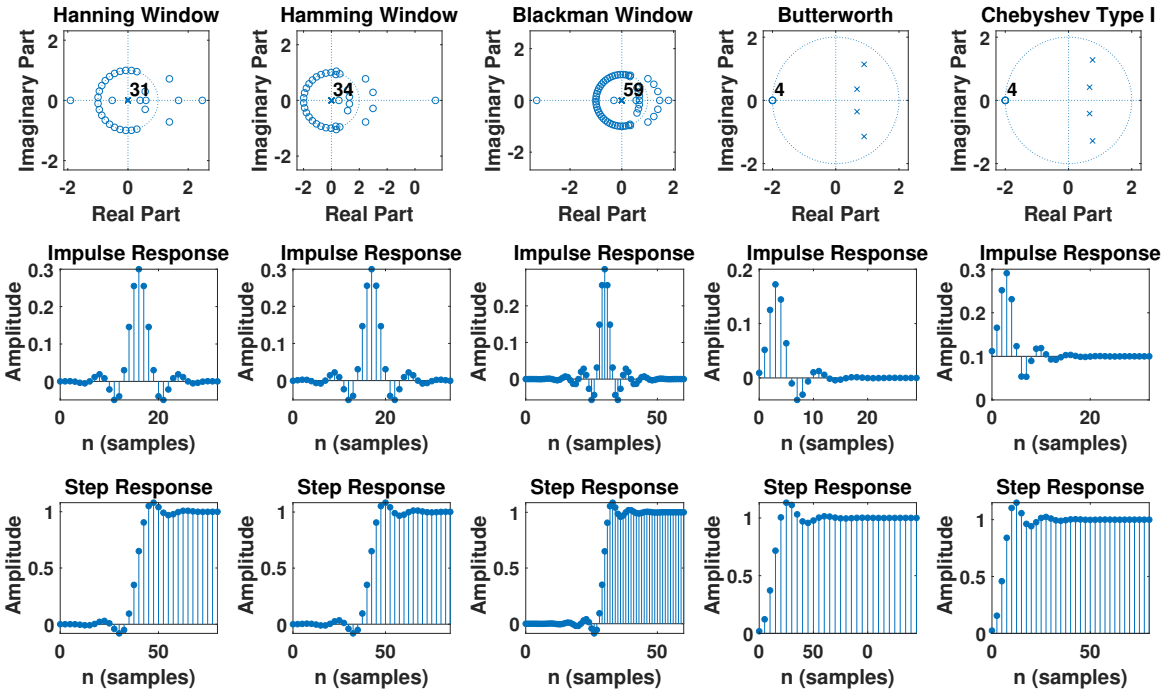


Figure F1: Design FIR and IIR filter zero-pole plot, impulse, and step response

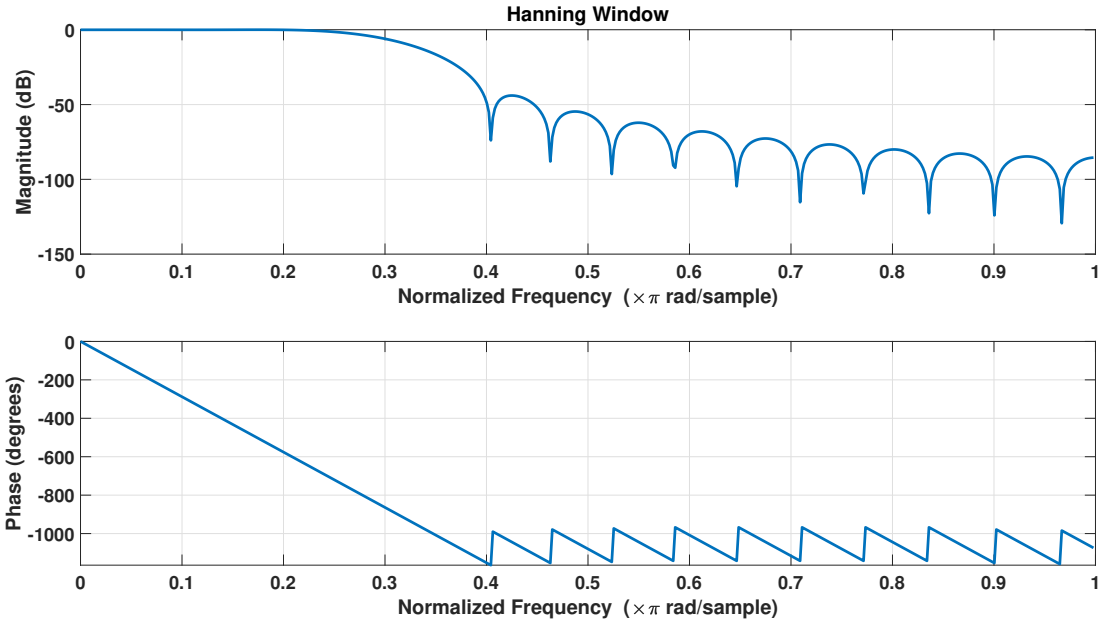


Figure F2: Designed Hanning filter shape

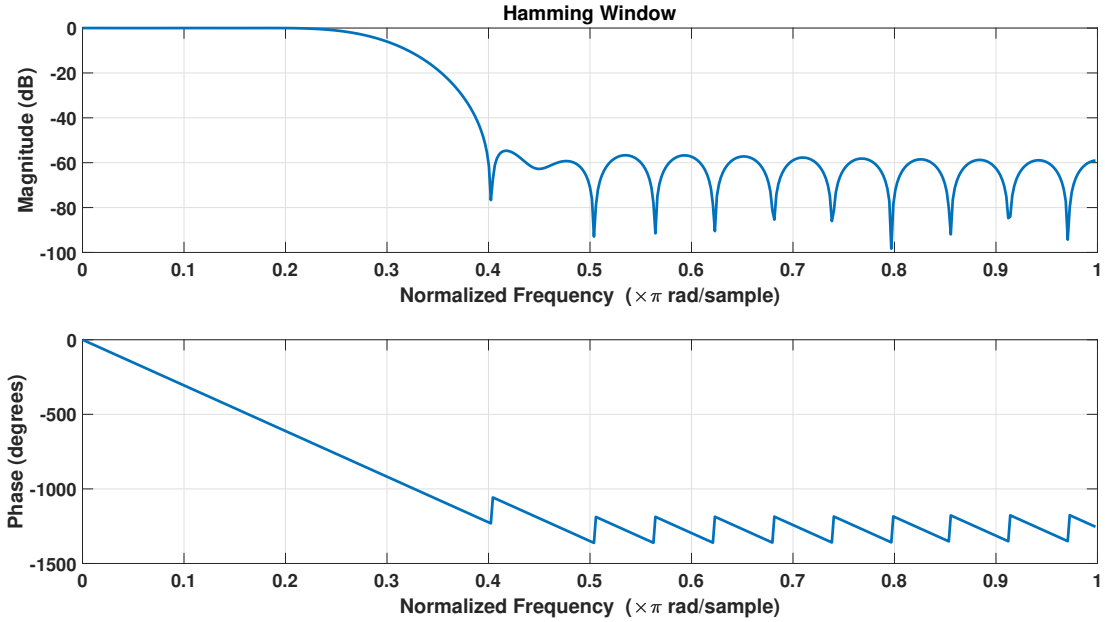


Figure F3: Designed Hamming filter shape

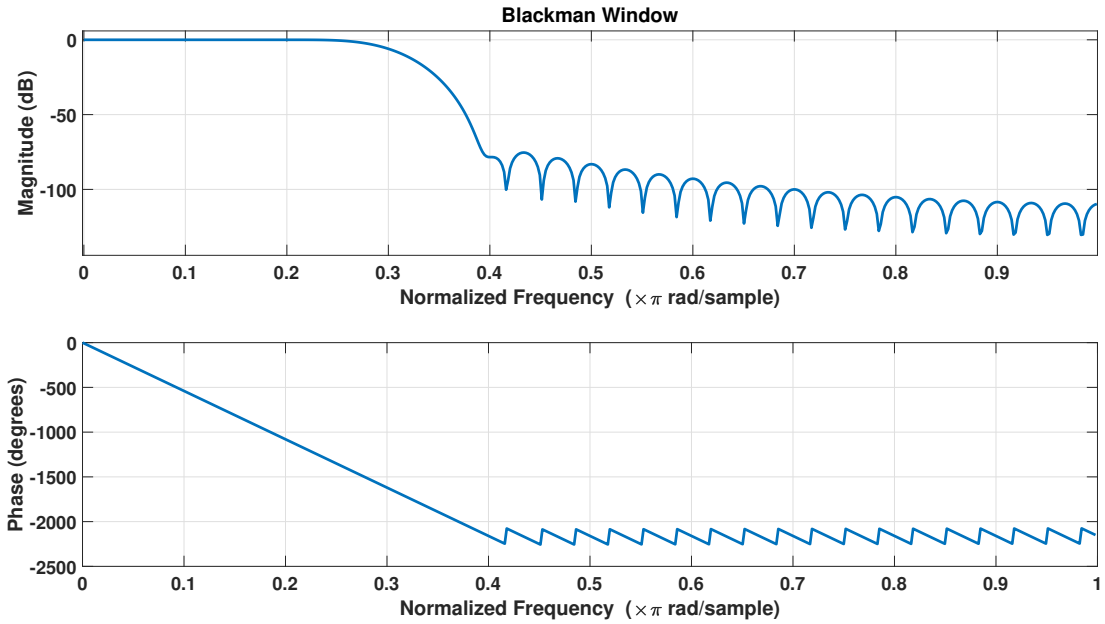


Figure F4: Designed blackman filter shape

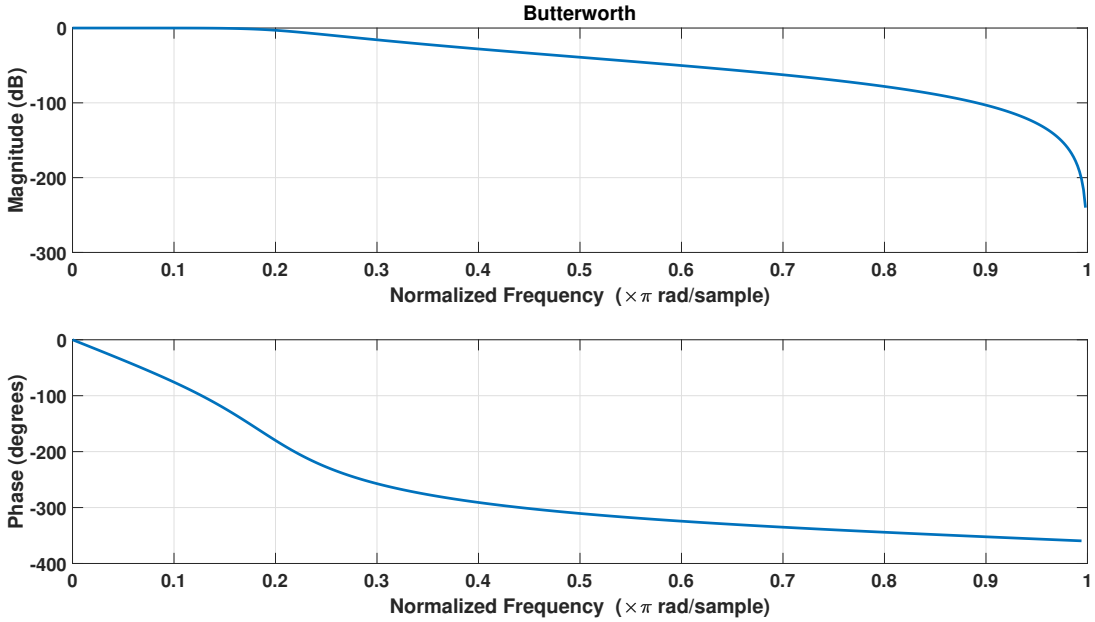


Figure F5: Designed butterworth filter shape

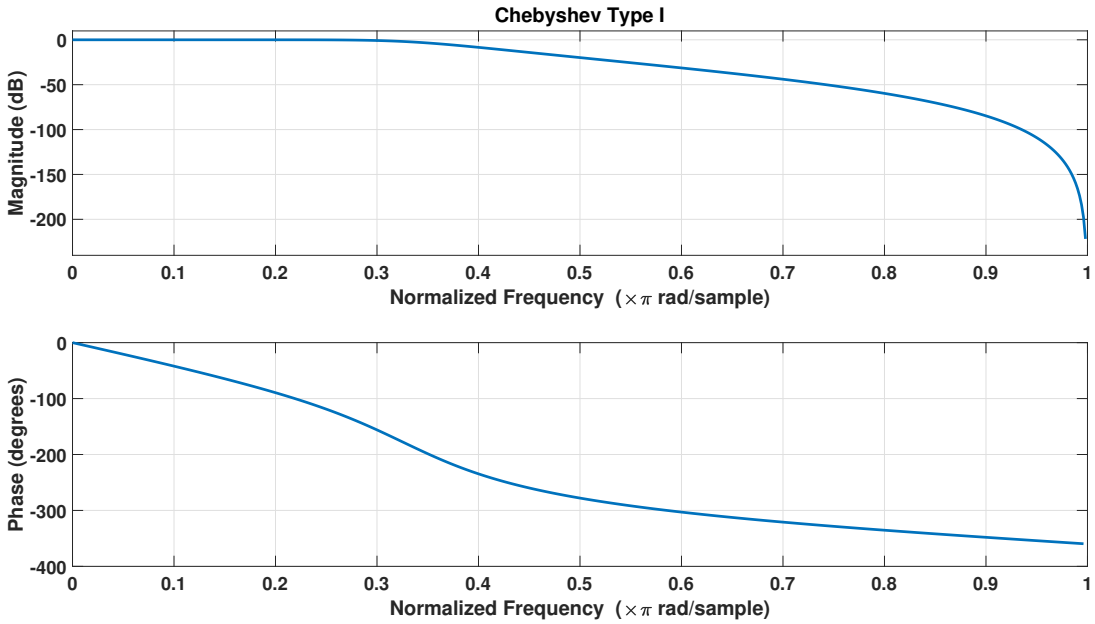


Figure F6: Designed Chebyshev filter shape

Appendix G: MATLAB codes

G1. Raw data, Filtered Plot, and plot configuration Matlab Code

```
clc
clear all
close all

%% Import Data
s1 = importdata('S1.mat');
s2 = importdata('S2.mat');

%% Data classification

ctr = input('Acceleration  type (1= Acceleration, 2 = Free acceleration) =      ');

    if ctr ==1
        n1 = 3; n2 = 4; n3 = 5;
    else
        n1 = 6; n2 = 7; n3 = 8;
    end
A_x_s1 = s1(:,n1);
A_y_s1 = s1(:,n2);
A_z_s1 = s1(:,n3);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A_x_s2 = s2(:,n1);
A_y_s2 = s2(:,n2);
A_z_s2 = s2(:,n3);

%% Finding Time
Fs = 100;             % Sampling frequency
T = 1/Fs;             % Sampling period
L = size(A_z_s1,1);             % Length of signal
t = (0:L-1)*T;        % Time vector
f = Fs*(0:(L/2))/L;

%% plot - time domain
h1= figure(1),set(gcf, 'Position', [0, 100, 1000, 550]);
subplot(3,2,1),plot(t,A_x_s1,'LineWidth', 1.5),legend('Sensor #1'),handlex9 = xlabel('time [s]');,
    handley9 = ylabel('a_x [m/s^2]');
xlim([0 35])
subplot(3,2,2),plot(t,A_x_s2,'LineWidth', 1.5),legend('Sensor #2'),handlex9 = xlabel('time [s]');,
    handley9 = ylabel('a_x [m/s^2]');
xlim([0 35])
subplot(3,2,3),plot(t,A_y_s1,'LineWidth', 1.5),legend('Sensor #1'),handlex9 = xlabel('time [s]');,
    handley9 = ylabel('a_y [m/s^2]');
xlim([0 35])
subplot(3,2,4),plot(t,A_y_s2,'LineWidth', 1.5),legend('Sensor #2'),handlex9 = xlabel('time [s]');,
    handley9 = ylabel('a_y [m/s^2]');
xlim([0 35])
subplot(3,2,5),plot(t,A_z_s1,'LineWidth', 1.5),legend('Sensor #1'),handlex9 = xlabel('time [s]');,
    handley9 = ylabel('a_z [m/s^2]');
xlim([0 35])
subplot(3,2,6),plot(t,A_z_s2,'LineWidth', 1.5),legend('Sensor #2'),handlex9 = xlabel('time [s]');,
```

```
        handley9 = ylabel('a_z [m/s^2]');
xlim([0 35])
%% Freq Domain
A_x_s1f = fft(A_x_s1);
P2A_x_s1f = abs(A_x_s1f/L);
P1A_x_s1f = P2A_x_s1f(1:L/2+1);
P1A_x_s1f(2:end-1) = 2*P1A_x_s1f(2:end-1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A_y_s1f = fft(A_y_s1);
P2A_y_s1f = abs(A_y_s1f/L);
P1A_y_s1f = P2A_y_s1f(1:L/2+1);
P1A_y_s1f(2:end-1) = 2*P1A_y_s1f(2:end-1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A_z_s1f = fft(abs(A_z_s1));
P2A_z_s1f = abs(A_z_s1f/L);
P1A_z_s1f = P2A_z_s1f(1:L/2+1);
P1A_z_s1f(2:end-1) = 2*P1A_z_s1f(2:end-1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A_x_s2f = fft(A_x_s2);
P2A_x_s2f = abs(A_x_s2f/L);
P1A_x_s2f = P2A_x_s2f(1:L/2+1);
P1A_x_s2f(2:end-1) = 2*P1A_x_s2f(2:end-1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A_y_s2f = fft(A_y_s2);
P2A_y_s2f = abs(A_y_s2f/L);
P1A_y_s2f = P2A_y_s2f(1:L/2+1);
P1A_y_s2f(2:end-1) = 2*P1A_y_s2f(2:end-1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A_z_s2f = fft(A_z_s2);
P2A_z_s2f = abs(A_z_s2f/L);
P1A_z_s2f = P2A_z_s2f(1:L/2+1);
P1A_z_s2f(2:end-1) = 2*P1A_z_s2f(2:end-1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h1= figure(2);
set(gcf, 'Position', [0, 100, 1000, 550]);
subplot(3,1,1),plot(f,P1A_x_s1f,f,P1A_x_s2f,'LineWidth', 1.5),title('X direction'),legend('Sensor #1',
    'Sensor #2'),handlex9 = xlabel('f (Hz)');, handley9 = ylabel('|P1(f)|');
subplot(3,1,2),plot(f,P1A_y_s1f,f,P1A_y_s2f,'LineWidth', 1.5),title('Y direction'),legend('Sensor #1',
    'Sensor #2'),handlex9 = xlabel('f (Hz)');, handley9 = ylabel('|P1(f)|');
subplot(3,1,3),plot(f,P1A_z_s1f,f,P1A_z_s2f,'LineWidth', 1.5),title('Z direction'),legend('Sensor #1',
    'Sensor #2'),handlex9 = xlabel('f (Hz)');, handley9 = ylabel('|P1(f)|');


%% Filter Design
fs = 100;

%%%%% FIR %%%%
PB = 10; % Pass Band Edge (Hz)
SB = 20; % Stop Band Edge (Hz)

TW = (SB-PB);
f1 = PB+(TW/2);
omega = 2*pi*(f1/fs);


%%%%%%%% Hanning Window %%%%%%%%

Nhann = 3.32*(fs/TW); % Hanning Window
```

```
Nhann = round(Nhann);

if rem(Nhann,2)==0
 Nhann = Nhann+1;
end

n_hann = [-(Nhann-1)/2:(Nhann-1)/2];
h_hann1 = (omega/pi).*sinc((omega/pi)*n_hann);

w_hann = hann(Nhann)';
h_hann = h_hann1.*w_hann;

b_hann = h_hann;
a_hann = 1;



A_x_s1_hnw = filtfilt(b_hann,a_hann,A_x_s1);
A_y_s1_hnw = filtfilt(b_hann,a_hann,A_y_s1);
A_z_s1_hnw = filtfilt(b_hann,a_hann,A_z_s1);

A_x_s2_hnw = filtfilt(b_hann,a_hann,A_x_s2);
A_y_s2_hnw = filtfilt(b_hann,a_hann,A_y_s2);
A_z_s2_hnw = filtfilt(b_hann,a_hann,A_z_s2);



%%%%%%%%% Hamming Window %%%%%%%%%
Nhamm = 3.44*(fs/TW); % Hamming Window
Nhamm = round(Nhamm);

if rem(Nhamm,2)==0
 Nhamm = Nhamm+1;
end

n_hamm = [-(Nhamm-1)/2:(Nhamm-1)/2];
h_hamm1 = (omega/pi).*sinc((omega/pi)*n_hamm);

w_hamm = hamming(Nhamm)';
h_hamm = h_hamm1.*w_hamm;

b_hamm = h_hamm;
a_hamm = 1;

A_x_s1_hmw = filtfilt(b_hamm,a_hamm,A_x_s1);
A_y_s1_hmw = filtfilt(b_hamm,a_hamm,A_y_s1);
A_z_s1_hmw = filtfilt(b_hamm,a_hamm,A_z_s1);

A_x_s2_hmw = filtfilt(b_hamm,a_hamm,A_x_s2);
A_y_s2_hmw = filtfilt(b_hamm,a_hamm,A_y_s2);
A_z_s2_hmw = filtfilt(b_hamm,a_hamm,A_z_s2);


%%%%%%%%% Blackman Window %%%%%%%%%
Nbl = 5.98*(fs/TW); % Blackman Window
Nbl = round(Nbl);

```

```
158 if rem(Nbl,2)==0
159  Nbl = Nbl+1;
160 end
161
162 n_bl = [-(Nbl-1)/2:(Nbl-1)/2];
163 h_bl1 = (omega/pi).*sinc((omega/pi)*n_bl);
164
165 w_bl = blackman(Nbl)';
166 h_bl = h_bl1.*w_bl;
167
168 b_bl = h_bl;
169 a_bl = 1;
170
171 A_x_s1_bw = filtfilt(b_bl,a_bl,A_x_s1);
172 A_y_s1_bw = filtfilt(b_bl,a_bl,A_y_s1);
173 A_z_s1_bw = filtfilt(b_bl,a_bl,A_z_s1);
174
175 A_x_s2_bw = filtfilt(b_bl,a_bl,A_x_s2);
176 A_y_s2_bw = filtfilt(b_bl,a_bl,A_y_s2);
177 A_z_s2_bw = filtfilt(b_bl,a_bl,A_z_s2);
178
179 %%% IIR Filter %%%%
180
181 %%%%%%%% Butterworth %%%%%%%%
182 fc = 10; % cut-off freq
183 fs = 100; % sampling freq
184 n_bw = 4; % order of filter
185 [b_bw,a_bw] = butter(n_bw,fc/(fs/2));
186
187 A_x_s1_but = filtfilt(b_bw,a_bw,A_x_s1);
188 A_y_s1_but = filtfilt(b_bw,a_bw,A_y_s1);
189 A_z_s1_but = filtfilt(b_bw,a_bw,A_z_s1);
190
191 A_x_s2_but = filtfilt(b_bw,a_bw,A_x_s2);
192 A_y_s2_but = filtfilt(b_bw,a_bw,A_y_s2);
193 A_z_s2_but = filtfilt(b_bw,a_bw,A_z_s2);
194
195
196 %%%%%%%% Chebyshev Type I %%%%%%%%
197
198 n_ch = 4; % Chebyshev Type I order
199 fp_ch = 10; % Pass band edge frequency (Hz)
200 Rp_ch = 0.001; % Pass band ripple (dB)
201 wp_ch = (2*pi*fp_ch)/(fs*pi);
202 [b_ch,a_ch] = cheby1(n_ch,Rp_ch,wp_ch);
203
204 A_x_s1_ch = filtfilt(b_ch,a_ch,A_x_s1);
205 A_y_s1_ch = filtfilt(b_ch,a_ch,A_y_s1);
206 A_z_s1_ch = filtfilt(b_ch,a_ch,A_z_s1);
207
208 A_x_s2_ch = filtfilt(b_ch,a_ch,A_x_s2);
209 A_y_s2_ch = filtfilt(b_ch,a_ch,A_y_s2);
210 A_z_s2_ch = filtfilt(b_ch,a_ch,A_z_s2);
211
212
213 %%%%%%%% Raw Data vs Filtered Data %%%%%%%%
214
215 %% FIR results
```

```
figure(3)
set(gcf, 'Position', [0, 100, 1000, 550]);
subplot(2,3,1),plot(t, A_x_s1, t,A_x_s1_hnw, t,A_x_s1_hmw, t,A_x_s1_bw,'LineWidth', 1.5),title('X
    direction - sensor #1'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_x [m/s^2]');
xlim([0 35])
subplot(2,3,2),plot(t, A_y_s1, t,A_y_s1_hnw, t,A_y_s1_hmw, t,A_y_s1_bw,'LineWidth', 1.5),title('y
    direction - sensor #1'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_y [m/s^2]');
xlim([0 35])
subplot(2,3,3),plot(t, A_z_s1, t,A_z_s1_hnw, t,A_z_s1_hmw, t,A_z_s1_bw,'LineWidth', 1.5),title('z
    direction - sensor #1'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_z [m/s^2]');
xlim([0 35])
subplot(2,3,4),plot(t, A_x_s1, t,A_x_s1_hnw, t,A_x_s1_hmw, t,A_x_s1_bw,'LineWidth', 1.5),title('X
    direction - sensor #1'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_x [m/s^2]');
xlim([0 35])
subplot(2,3,5),plot(t, A_y_s1, t,A_y_s1_hnw, t,A_y_s1_hmw, t,A_y_s1_bw,'LineWidth', 1.5),title('y
    direction - sensor #1'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_y [m/s^2]');
xlim([0 35])
subplot(2,3,6),plot(t, A_z_s1, t,A_z_s1_hnw, t,A_z_s1_hmw, t,A_z_s1_bw,'LineWidth', 1.5),title('z
    direction - sensor #1'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_z [m/s^2]');
xlim([0 35])

figure(4)
set(gcf, 'Position', [0, 100, 1000, 550]);
subplot(2,3,1),plot(t, A_x_s2, t,A_x_s2_hnw, t,A_x_s2_hmw, t,A_x_s2_bw,'LineWidth', 1.5),title('X
    direction - sensor #2'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_x [m/s^2]');
xlim([0 35])
subplot(2,3,2),plot(t, A_y_s2, t,A_y_s2_hnw, t,A_y_s2_hmw, t,A_y_s2_bw,'LineWidth', 1.5),title('y
    direction - sensor #2'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_y [m/s^2]');
xlim([0 35])
subplot(2,3,3),plot(t, A_z_s2, t,A_z_s2_hnw, t,A_z_s2_hmw, t,A_z_s2_bw,'LineWidth', 1.5),title('z
    direction - sensor #2'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_z [m/s^2]');
xlim([0 35])
subplot(2,3,4),plot(t, A_x_s2, t,A_x_s2_hnw, t,A_x_s2_hmw, t,A_x_s2_bw,'LineWidth', 1.5),title('X
    direction - sensor #2'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_x [m/s^2]');
xlim([0 35])
subplot(2,3,5),plot(t, A_y_s2, t,A_y_s2_hnw, t,A_y_s2_hmw, t,A_y_s2_bw,'LineWidth', 1.5),title('y
    direction - sensor #2'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_y [m/s^2]');
xlim([0 35])
subplot(2,3,6),plot(t, A_z_s2, t,A_z_s2_hnw, t,A_z_s2_hmw, t,A_z_s2_bw,'LineWidth', 1.5),title('z
    direction - sensor #2'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_z [m/s^2]');
xlim([0 35])


%% FIR results For slides
% close all
%
```

```
% figure(1)
% set(gcf, 'Position', [0, 100, 1000, 550]);
% subplot(2,1,1),plot(t, A_x_s1, t,A_x_s1_hnw, t,A_x_s1_hmw, t,A_x_s1_bw,'LineWidth', 1.5),title('X
    direction - sensor #1'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_x [m/s^2]');
% xlim([10 30])
% subplot(2,1,2),plot(t, A_x_s1, t,A_x_s1_hnw, t,A_x_s1_hmw, t,A_x_s1_bw,'LineWidth', 1.5),title('X
    direction - sensor #1'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_x [m/s^2]');
% xlim([10 30])
%
% figure(2)
% set(gcf, 'Position', [0, 100, 1000, 550]);
% subplot(2,1,1),plot(t, A_y_s1, t,A_y_s1_hnw, t,A_y_s1_hmw, t,A_y_s1_bw,'LineWidth', 1.5),title('y
    direction - sensor #1'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_y [m/s^2]');
% xlim([10 30])
% subplot(2,1,2),plot(t, A_y_s1, t,A_y_s1_hnw, t,A_y_s1_hmw, t,A_y_s1_bw,'LineWidth', 1.5),title('y
    direction - sensor #1'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_y [m/s^2]');
% xlim([10 30])
%
% figure(3)
% set(gcf, 'Position', [0, 100, 1000, 550]);
% subplot(2,1,1),plot(t, A_z_s1, t,A_z_s1_hnw, t,A_z_s1_hmw, t,A_z_s1_bw,'LineWidth', 1.5),title('z
    direction - sensor #1'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_z [m/s^2]');
% xlim([10 30])
% subplot(2,1,2),plot(t, A_z_s1, t,A_z_s1_hnw, t,A_z_s1_hmw, t,A_z_s1_bw,'LineWidth', 1.5),title('z
    direction - sensor #1'),legend('Raw Data', 'Hanning Window','Hamming Window','Blackman Window'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('a_z [m/s^2]');
% xlim([10 30])
%% IIR Results

figure(5)
set(gcf, 'Position', [0, 100, 1000, 550]);
subplot(2,3,1),plot(t, A_x_s1, t,A_x_s1_but, t,A_x_s1_ch,'LineWidth', 1.5),title('X direction - sensor
    #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');, handley9 =
    ylabel('a_x [m/s^2]');
xlim([10 30])
subplot(2,3,2),plot(t, A_y_s1, t,A_y_s1_but, t,A_y_s1_ch,'LineWidth', 1.5),title('y direction - sensor
    #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');, handley9 =
    ylabel('a_y [m/s^2]');
xlim([10 30])
subplot(2,3,3),plot(t, A_z_s1, t,A_z_s1_but, t,A_z_s1_ch,'LineWidth', 1.5),title('z direction - sensor
    #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');, handley9 =
    ylabel('a_z [m/s^2]');
xlim([10 30])
subplot(2,3,4),plot(t, A_x_s1, t,A_x_s1_but, t,A_x_s1_ch,'LineWidth', 1.5),title('X direction - sensor
    #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');, handley9 =
    ylabel('a_x [m/s^2]');
xlim([10 30])
subplot(2,3,5),plot(t, A_y_s1, t,A_y_s1_but, t,A_y_s1_ch,'LineWidth', 1.5),title('y direction - sensor
    #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');, handley9 =
    ylabel('a_y [m/s^2]');
xlim([10 30])
subplot(2,3,6),plot(t, A_z_s1, t,A_z_s1_but, t,A_z_s1_ch,'LineWidth', 1.5),title('z direction - sensor
    #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');, handley9 =
```

```
        ylabel('a_z [m/s^2]');
285 xlim([10 30])
286
287 figure(6)
288 set(gcf, 'Position', [0, 100, 1000, 550]);
289 subplot(2,3,1),plot(t, A_x_s2, t,A_x_s2_but, t,A_x_s2_ch,'LineWidth', 1.5),title('X direction - sensor
        #2'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');, handley9 =
        ylabel('a_x [m/s^2]');
290 xlim([0 35])
291 subplot(2,3,2),plot(t, A_y_s2, t,A_y_s2_but, t,A_y_s2_ch,'LineWidth', 1.5),title('y direction - sensor
        #2'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');, handley9 =
        ylabel('a_y [m/s^2]');
292 xlim([0 35])
293 subplot(2,3,3),plot(t, A_z_s2, t,A_z_s2_but, t,A_z_s2_ch,'LineWidth', 1.5),title('z direction - sensor
        #2'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');, handley9 =
        ylabel('a_z [m/s^2]');
294 xlim([0 35])
295 subplot(2,3,4),plot(t, A_x_s2, t,A_x_s2_but, t,A_x_s2_ch,'LineWidth', 1.5),title('X direction - sensor
        #2'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');, handley9 =
        ylabel('a_x [m/s^2]');
296 xlim([0 35])
297 subplot(2,3,5),plot(t, A_y_s2, t,A_y_s2_but, t,A_y_s2_ch,'LineWidth', 1.5),title('y direction - sensor
        #2'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');, handley9 =
        ylabel('a_y [m/s^2]');
298 xlim([0 35])
299 subplot(2,3,6),plot(t, A_z_s2, t,A_z_s2_but, t,A_z_s2_ch,'LineWidth', 1.5),title('z direction - sensor
        #2'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');, handley9 =
        ylabel('a_z [m/s^2]');
300 xlim([0 35])
301 %% IIR results For slides
302 % close all
303 % figure(1)
304 % set(gcf, 'Position', [0, 100, 1000, 550]);
305 % subplot(2,1,1),plot(t, A_x_s1, t,A_x_s1_but, t,A_x_s1_ch,'LineWidth', 1.5),title('X direction -
        sensor #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');,
        handley9 = ylabel('a_x [m/s^2]');
306 % xlim([10 30])
307 % subplot(2,1,2),plot(t, A_x_s1, t,A_x_s1_but, t,A_x_s1_ch,'LineWidth', 1.5),title('X direction -
        sensor #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');,
        handley9 = ylabel('a_x [m/s^2]');
308 % xlim([10 30])
309 %
310 % figure(2)
311 % set(gcf, 'Position', [0, 100, 1000, 550]);
312 % subplot(2,1,1),plot(t, A_y_s1, t,A_y_s1_but, t,A_y_s1_ch,'LineWidth', 1.5),title('y direction -
        sensor #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');,
        handley9 = ylabel('a_y [m/s^2]');
313 % xlim([10 30])
314 % subplot(2,1,2),plot(t, A_y_s1, t,A_y_s1_but, t,A_y_s1_ch,'LineWidth', 1.5),title('y direction -
        sensor #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');,
        handley9 = ylabel('a_y [m/s^2]');
315 % xlim([10 30])
316 %
317 % figure(3)
318 % set(gcf, 'Position', [0, 100, 1000, 550]);
319 % subplot(2,1,1),plot(t, A_z_s1, t,A_z_s1_but, t,A_z_s1_ch,'LineWidth', 1.5),title('z direction -
        sensor #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');,
        handley9 = ylabel('a_z [m/s^2]');
```

```
% xlim([10 30])
% subplot(2,1,2),plot(t, A_z_s1, t,A_z_s1_but, t,A_z_s1_ch,'LineWidth', 1.5),title('z direction -
    sensor #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),handlex9 = xlabel('t [s]');,
    handley9 = ylabel('a_z [m/s^2]');
% xlim([10 30])


%% Velocity Plot

%%%%%%%% Data input for integral %%%%%%%%
data1 = [A_x_s1 A_y_s1 A_z_s1]; %raw data sensor #1
data2 = [A_x_s2 A_y_s2 A_z_s2]; %raw data sensor #1
data1_but = [A_x_s1_but A_y_s1_but A_z_s1_but]; %Filtered Butterworth  sensor #1
data2_but = [A_x_s2_but A_y_s2_but A_z_s2_but]; %Filtered Butterworth  sensor #2

data1_ch = [A_x_s1_ch A_y_s1_ch A_z_s1_ch]; %Filtered Chebyshev Type I sensor #1
data2_ch = [A_x_s2_ch A_y_s2_ch A_z_s2_ch]; %Filtered Chebyshev Type I sensor #2

data1_hnw = [A_x_s1_hnw A_y_s1_hnw A_z_s1_hnw]; %Filtered hanning sensor #1
data2_hnw = [A_x_s2_hnw A_y_s2_hnw A_z_s2_hnw]; %Filtered hanning sensor #1

data1_hmw = [A_x_s1_hmw A_y_s1_hmw A_z_s1_hmw]; %Filtered hamming sensor #1
data2_hmw = [A_x_s2_hmw A_y_s2_hmw A_z_s2_hmw]; %Filtered hamming sensor #1

data1_bw = [A_x_s1_bw A_y_s1_bw A_z_s1_bw]; %Filtered Blackman sensor #1
data2_bw = [A_x_s2_bw A_y_s2_bw A_z_s2_bw]; %Filtered Blackman sensor #1

%%%%%%%% Velocity %%%%%%%%
%x
[time,vel_x]=velfinder(data1,data2,1);
[~,vel_x_but]=velfinder(data1_but,data2_but,1);
[~,vel_x_ch]=velfinder(data1_ch,data2_ch,1);
[~,vel_x_hnw]=velfinder(data1_hnw,data2_hnw,1);
[~,vel_x_hmw]=velfinder(data1_hmw,data2_hmw,1);
[~,vel_x_bw]=velfinder(data1_bw,data2_bw,1);

%y
[~,vel_y]=velfinder(data1,data2,2);
[~,vel_y_but]=velfinder(data1_but,data2_but,2);
[~,vel_y_ch]=velfinder(data1_ch,data2_ch,2);
[~,vel_y_hnw]=velfinder(data1_hnw,data2_hnw,2);
[~,vel_y_hmw]=velfinder(data1_hmw,data2_hmw,2);
[~,vel_y_bw]=velfinder(data1_bw,data2_bw,2);

%z
[~,vel_z]=velfinder(data1,data2,3);
[~,vel_z_but]=velfinder(data1_but,data2_but,3);
[~,vel_z_ch]=velfinder(data1_ch,data2_ch,3);
[~,vel_z_hnw]=velfinder(data1_hnw,data2_hnw,3);
[~,vel_z_hmw]=velfinder(data1_hmw,data2_hmw,3);
[~,vel_z_bw]=velfinder(data1_bw,data2_bw,3);


%%%%%%%% position %%%%%%%%
%x
[time_x,pos_x]=posfinder(data1,data2,1);
[~,pos_x_but]=posfinder(data1_but,data2_but,1);
[~,pos_x_ch]=posfinder(data1_ch,data2_ch,1);
```

```
[~,pos_x_hnw]=posfinder(data1_hnw,data2_hnw,1);
[~,pos_x_hmw]=posfinder(data1_hmw,data2_hmw,1);
[~,pos_x_bw]=posfinder(data1_bw,data2_bw,1);

%y
[~,pos_y]=posfinder(data1,data2,2);
[~,pos_y_but]=posfinder(data1_but,data2_but,2);
[~,pos_y_ch]=posfinder(data1_ch,data2_ch,2);
[~,pos_y_hnw]=posfinder(data1_hnw,data2_hnw,2);
[~,pos_y_hmw]=posfinder(data1_hmw,data2_hmw,2);
[~,pos_y_bw]=posfinder(data1_bw,data2_bw,2);

%z
[~,pos_z]=posfinder(data1,data2,3);
[~,pos_z_but]=posfinder(data1_but,data2_but,3);
[~,pos_z_ch]=posfinder(data1_ch,data2_ch,3);
[~,pos_z_hnw]=posfinder(data1_hnw,data2_hnw,3);
[~,pos_z_hmw]=posfinder(data1_hmw,data2_hmw,3);
[~,pos_z_bw]=posfinder(data1_bw,data2_bw,3);


%% Velocity

figure(7)
set(gcf, 'Position', [0, 100, 1000, 550]);
subplot(3,1,1),plot(time, vel_x(:,1), time, vel_x_hnw(:,1), time, vel_x_hmw(:,1), time, vel_x_bw(:,1),'
    LineWidth', 1.5),title('Velocity in X direction - sensor #1'),legend('Raw Data', 'Hanning Window',
    'Hamming Window','Blackman Window'),handlex9 = xlabel('t [s]');, handley9 = ylabel('v_x [m/s]');
xlim([0 35])
subplot(3,1,2),plot(time, vel_y(:,1), time, vel_y_hnw(:,1), time, vel_y_hmw(:,1), time, vel_y_bw(:,1),'
    LineWidth', 1.5),title('Velocity in y direction - sensor #1'),legend('Raw Data', 'Hanning Window',
    'Hamming Window','Blackman Window'),handlex9 = xlabel('t [s]');, handley9 = ylabel('v_y [m/s]');
xlim([0 35])
subplot(3,1,3),plot(time, vel_z(:,1), time, vel_z_hnw(:,1), time, vel_z_hmw(:,1), time, vel_z_bw(:,1),'
    LineWidth', 1.5),title('Velocity in z direction - sensor #1'),legend('Raw Data', 'Hanning Window',
    'Hamming Window','Blackman Window'),handlex9 = xlabel('t [s]');, handley9 = ylabel('v_z [m/s]');
xlim([0 35])

figure(8)
set(gcf, 'Position', [0, 100, 1000, 550]);
subplot(3,1,1),plot(time, vel_x(:,2), time, vel_x_hnw(:,2), time, vel_x_hmw(:,2), time, vel_x_bw(:,2),'
    LineWidth', 1.5),title('Velocity in X direction - sensor #2'),legend('Raw Data', 'Hanning Window',
    'Hamming Window','Blackman Window'),handlex9 = xlabel('t [s]');, handley9 = ylabel('v_x [m/s]');
xlim([0 35])
subplot(3,1,2),plot(time, vel_y(:,2), time, vel_y_hnw(:,2), time, vel_y_hmw(:,2), time, vel_y_bw(:,2),'
    LineWidth', 1.5),title('Velocity in y direction - sensor #2'),legend('Raw Data', 'Hanning Window',
    'Hamming Window','Blackman Window'),handlex9 = xlabel('t [s]');, handley9 = ylabel('v_y [m/s]');
xlim([0 35])
subplot(3,1,3),plot(time, vel_z(:,2), time, vel_z_hnw(:,2), time, vel_z_hmw(:,2), time, vel_z_bw(:,2),'
    LineWidth', 1.5),title('Velocity in z direction - sensor #2'),legend('Raw Data', 'Hanning Window',
    'Hamming Window','Blackman Window'),handlex9 = xlabel('t [s]');, handley9 = ylabel('v_z [m/s]');
xlim([0 35])

figure(9)
set(gcf, 'Position', [0, 100, 1000, 550]);
subplot(3,1,1),plot(time, vel_x(:,1), time, vel_x_but(:,1), time, vel_x_ch(:,1),'LineWidth', 1.5),title
    ('Velocity in X direction - sensor #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('v_x [m/s]');
```

```
xlim([0 35])
subplot(3,1,2),plot(time, vel_y(:,1), time, vel_y_but(:,1), time, vel_y_ch(:,1),'LineWidth', 1.5),title
    ('Velocity in y direction - sensor #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('v_y [m/s]');
xlim([0 35])
subplot(3,1,3),plot(time, vel_z(:,1), time, vel_z_but(:,1), time, vel_z_ch(:,1),'LineWidth', 1.5),title
    ('Velocity in z direction - sensor #1'),legend('Raw Data','Butterworth','Chebyshev Type I'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('v_z [m/s]');
xlim([0 35])

figure(10)
set(gcf, 'Position', [0, 100, 1000, 550]);
subplot(3,1,1),plot(time, vel_x(:,2), time, vel_x_but(:,2), time, vel_x_ch(:,2),'LineWidth', 1.5),title
    ('Velocity in X direction - sensor #2'),legend('Raw Data','Butterworth','Chebyshev Type I'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('v_x [m/s]');
xlim([0 35])
subplot(3,1,2),plot(time, vel_y(:,2), time, vel_y_but(:,2), time, vel_y_ch(:,2),'LineWidth', 1.5),title
    ('Velocity in y direction - sensor #2'),legend('Raw Data','Butterworth','Chebyshev Type I'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('v_y [m/s]');
xlim([0 35])
subplot(3,1,3),plot(time, vel_z(:,2), time, vel_z_but(:,2), time, vel_z_ch(:,2),'LineWidth', 1.5),title
    ('Velocity in z direction - sensor #2'),legend('Raw Data','Butterworth','Chebyshev Type I'),
    handlex9 = xlabel('t [s]');, handley9 = ylabel('v_z [m/s]');
xlim([0 35])

%%  Position results
figure(11)
set(gcf, 'Position', [0, 100, 1000, 550]);
subplot(3,1,1),plot(time_x, pos_x(:,1), time_x, pos_x_hnw(:,1), time_x, pos_x_hmw(:,1), time_x,
    pos_x_bw(:,1),'LineWidth', 1.5),title('Displacement in X direction - sensor #1'),legend('Raw Data'
    , 'Hanning Window','Hamming Window','Blackman Window'),handlex9 = xlabel('t [s]');, handley9 =
    ylabel('x [m]');
xlim([0 35])
subplot(3,1,2),plot(time_x, pos_y(:,1), time_x, pos_y_hnw(:,1), time_x, pos_y_hmw(:,1), time_x,
    pos_y_bw(:,1),'LineWidth', 1.5),title('Displacement in y direction - sensor #1'),legend('Raw Data'
    , 'Hanning Window','Hamming Window','Blackman Window'),handlex9 = xlabel('t [s]');, handley9 =
    ylabel('y [m]');
xlim([0 35])
subplot(3,1,3),plot(time_x, pos_z(:,1), time_x, pos_z_hnw(:,1), time_x, pos_z_hmw(:,1), time_x,
    pos_z_bw(:,1),'LineWidth', 1.5),title('Displacement in z direction - sensor #1'),legend('Raw Data'
    , 'Hanning Window','Hamming Window','Blackman Window'),handlex9 = xlabel('t [s]');, handley9 =
    ylabel('z [m]');
xlim([0 35])

figure(12)
set(gcf, 'Position', [0, 100, 1000, 550]);
subplot(3,1,1),plot(time_x, pos_x(:,2), time_x, pos_x_hnw(:,2), time_x, pos_x_hmw(:,2), time_x,
    pos_x_bw(:,2),'LineWidth', 1.5),title('Displacement in X direction - sensor #2'),legend('Raw Data'
    , 'Hanning Window','Hamming Window','Blackman Window'),handlex9 = xlabel('t [s]');, handley9 =
    ylabel('x [m]');
xlim([0 35])
subplot(3,1,2),plot(time_x, pos_y(:,2), time_x, pos_y_hnw(:,2), time_x, pos_y_hmw(:,2), time_x,
    pos_y_bw(:,2),'LineWidth', 1.5),title('Displacement in y direction - sensor #2'),legend('Raw Data'
    , 'Hanning Window','Hamming Window','Blackman Window'),handlex9 = xlabel('t [s]');, handley9 =
    ylabel('y [m]');
xlim([0 35])
subplot(3,1,3),plot(time_x, pos_z(:,2), time_x, pos_z_hnw(:,2), time_x, pos_z_hmw(:,2), time_x,
    pos_z_bw(:,2),'LineWidth', 1.5),title('Displacement in z direction - sensor #2'),legend('Raw Data'
```

```
    , 'Hanning Window','Hamming Window','Blackman Window'),handlex9 = xlabel('t [s]');, handley9 =
    ylabel('z [m]');
452
453
454 figure(13)
455 set(gcf, 'Position', [0, 100, 1000, 550]);
456 subplot(3,1,1),plot(time_x, pos_x(:,1), time_x, pos_x_but(:,1), time_x, pos_x_ch(:,1),'LineWidth', 1.5)
    ,title('Displacement in X direction - sensor #1'),legend('Raw Data','Butterworth','Chebyshev Type
    I'),handlex9 = xlabel('t [s]');, handley9 = ylabel('x [m]');
457 xlim([0 35])
458 subplot(3,1,2),plot(time_x, pos_y(:,1), time_x, pos_y_but(:,1), time_x, pos_y_ch(:,1),'LineWidth', 1.5)
    ,title('Displacement in y direction - sensor #1'),legend('Raw Data','Butterworth','Chebyshev Type
    I'),handlex9 = xlabel('t [s]');, handley9 = ylabel('y [m]');
459 xlim([0 35])
460 subplot(3,1,3),plot(time_x, pos_z(:,1), time_x, pos_z_but(:,1), time_x, pos_z_ch(:,1),'LineWidth', 1.5)
    ,title('Displacement in z direction - sensor #1'),legend('Raw Data','Butterworth','Chebyshev Type
    I'),handlex9 = xlabel('t [s]');, handley9 = ylabel('z [m]');
461 xlim([0 35])
462
463 figure(14)
464 set(gcf, 'Position', [0, 100, 1000, 550]);
465 subplot(3,1,1),plot(time_x, pos_x(:,2), time_x, pos_x_but(:,2), time_x, pos_x_ch(:,2),'LineWidth', 1.5)
    ,title('Displacement in X direction - sensor #2'),legend('Raw Data','Butterworth','Chebyshev Type
    I'),handlex9 = xlabel('t [s]');, handley9 = ylabel('x [m]');
466 xlim([0 35])
467 subplot(3,1,2),plot(time_x, pos_y(:,2), time_x, pos_y_but(:,2), time_x, pos_y_ch(:,2),'LineWidth', 1.5)
    ,title('Displacement in y direction - sensor #2'),legend('Raw Data','Butterworth','Chebyshev Type
    I'),handlex9 = xlabel('t [s]');, handley9 = ylabel('y [m]]');
468 xlim([0 35])
469 subplot(3,1,3),plot(time_x, pos_z(:,2), time_x, pos_z_but(:,2), time_x, pos_z_ch(:,2),'LineWidth', 1.5)
    ,title('Displacement in z direction - sensor #2'),legend('Raw Data','Butterworth','Chebyshev Type
    I'),handlex9 = xlabel('t [s]');, handley9 = ylabel('z [m]');
470 xlim([0 35])
471
472 %% Filter Config
473 figure(15)
474 set(gcf, 'Position', [0, 100, 1000, 550]);
475
476 % Pole-Zero map
477 subplot(3,5,1),zplane(b_hann,a_hann),title('Hanning Window');
478 subplot(3,5,2),zplane(b_hamm,a_hamm),title('Hamming Window');
479 subplot(3,5,3),zplane(b_bl,a_bl),title('Blackman Window');
480 subplot(3,5,4),zplane(b_bw,a_bw),title('Butterworth');
481 subplot(3,5,5),zplane(b_ch,a_ch),title('Chebyshev Type I');
482
483 % Impulse Response
484 subplot(3,5,6),impz(b_hann,a_hann),title('Hanning Window');
485 subplot(3,5,7),impz(b_hamm,a_hamm),title('Hamming Window');
486 subplot(3,5,8),impz(b_bl,a_bl),title('Blackman Window');
487 subplot(3,5,9),impz(b_bw,a_bw),title('Butterworth');
488 subplot(3,5,10),impz(b_ch,a_ch),title('Chebyshev Type I');
489
490 % Step Response
491 subplot(3,5,11),stepz(b_hann,a_hann),title('Hanning Window');
492 subplot(3,5,12),stepz(b_hamm,a_hamm),title('Hamming Window');
493 subplot(3,5,13),stepz(b_bl,a_bl),title('Blackman Window');
494 subplot(3,5,14),stepz(b_bw,a_bw),title('Butterworth');
495 subplot(3,5,15),stepz(b_ch,a_ch),title('Chebyshev Type I');
```

```
496
497 %% Filter Shape
498
499 % Hanning Window
500 figure(16)
501 set(gcf, 'Position', [10, 200, 1200, 800]);
502 freqz(b_hann,a_hann),title('Hanning Window');
503
504 % Hamming Window
505 figure(17)
506 set(gcf, 'Position', [10, 200, 1200, 800]);
507 freqz(b_hamm,a_hamm),title('Hamming Window');
508
509 % Blackman Window
510 figure(18)
511 set(gcf, 'Position', [10, 200, 1200, 800]);
512 freqz(b_bl,a_bl),title('Blackman Window');
513
514 % Butterworth
515 figure(19)
516 set(gcf, 'Position', [10, 200, 1200, 800]);
517 subplot(2,5,[4,9]),freqz(b_bw,a_bw),title('Butterworth');
518
519 % Chebyshev Type I
520 figure(20)
521 set(gcf, 'Position', [10, 200, 1200, 800]);
522 subplot(2,5,[5,10]),freqz(b_ch,a_ch),title('Chebyshev Type I');
```

G2. Function of Finding velocity

```
1  function [time,vel]=velfinder(data,data2,n)
2  time=(1:size(data,1))/100;
3  N(1).trdata=zeros(size(data));
4  N(1).trdata(1300:2390,:)=data(1300:2390,:);
5  N(2).trdata=zeros(size(data2));
6  N(2).trdata(1300:2390,:)=data2(1300:2390,:);
7  for i=1:2
8      for j=1:3
9  N(i).freeinGCS(:,j)=N(i).trdata(:,j);
10     end
11 end
12 for i=1:2
13 for j=1:3
14 N(i).Vel(:,j)=cumtrapz(time,N(i).freeinGCS(:,j));
15 end
16 end
17
18 for i=1:2
19 for j=1:3
20 N(i).Slope2(j)=(N(i).Vel(2390,j)-N(i).Vel(1300,j))/1090;
21 end
22 end
23 for i=1:2
24     for j=1:3
25         for k=1:size(data,1)
```

```
        if k>=1300 && k<=2390
N(i).ModVel(k,j)= N(i).Vel(k,j)-N(i).Slope2(j)*(k-1300)-N(i).Vel(1300,j);
        else
N(i).ModVel(k,j)= 0;
        end
        end
    end
end


for i=1:2
for j=1:3
N(i).Position(:,j)=cumtrapz(time,N(i).ModVel(:,j));
end
end
vel=[N(1).ModVel(:,n),N(2).ModVel(:,n)];
end
```

G3. Function of Finding Position

```
function [time,position, n]=posfinder(data,data2, n)
time=(1:size(data,1))/100;
N(1).trdata=zeros(size(data));
N(1).trdata(1300:2390,:)=data(1300:2390,:);
N(2).trdata=zeros(size(data2));
N(2).trdata(1300:2390,:)=data2(1300:2390,:);
for i=1:2
    for j=1:3
N(i).freeinGCS(:,j)=N(i).trdata(:,j);
    end
end
for i=1:2
for j=1:3
N(i).Vel(:,j)=cumtrapz(time,N(i).freeinGCS(:,j));
end
end

for i=1:2
for j=1:3
N(i).Slope2(j)=(N(i).Vel(2390,j)-N(i).Vel(1300,j))/1090;
end
end
for i=1:2
    for j=1:3
        for k=1:size(data,1)
        if k>=1300 && k<=2390
N(i).ModVel(k,j)= N(i).Vel(k,j)-N(i).Slope2(j)*(k-1300)-N(i).Vel(1300,j);
        else
N(i).ModVel(k,j)= 0;
        end
        end
    end
end


```

```
for i=1:2
for j=1:3
N(i).Position(:,j)=cumtrapz(time,N(i).ModVel(:,j));
end
end
position=[N(1).Position(:,n),N(2).Position(:,n)];
end
```

G4. Function of Finding Temporal Spatial Parameters

```

function [Lstridetime,Lstridelength,Lsteptime,Lsteplength, Rstridetime,Rstridelength,Rsteptime,
    Rsteplength, Rspeed,Lspeed]=TSPfinder(data,data2)
time=(1:size(data,1))/100;
N(1).trdata=zeros(size(data));
N(1).trdata(1300:2390,:)=data(1300:2390,:);
N(2).trdata=zeros(size(data2));
N(2).trdata(1300:2390,:)=data2(1300:2390,:);
for i=1:2
    for j=1:3
N(i).freeinGCS(:,j)=N(i).trdata(:,j);
    end
end
for i=1:2
for j=1:3
N(i).Vel(:,j)=cumtrapz(time,N(i).freeinGCS(:,j));
end
end

for i=1:2
for j=1:3
N(i).Slope2(j)=(N(i).Vel(2390,j)-N(i).Vel(1300,j))/1090;
end
end
for i=1:2
    for j=1:3
        for k=1:size(data,1)
        if k>=1300 && k<=2390
N(i).ModVel(k,j)= N(i).Vel(k,j)-N(i).Slope2(j)*(k-1300)-N(i).Vel(1300,j);
        else
N(i).ModVel(k,j)= 0;
        end
        end
    end
end


for i=1:2
for j=1:3
N(i).Position(:,j)=cumtrapz(time,N(i).ModVel(:,j));
end
end
%

%% Temporal Spatial Parameters
```

```
Vel=N(1).ModVel;
Vel2=N(2).ModVel;
Position=N(1).Position;
Position2=N(2).Position;
[PKSR,LOCSR]= findpeaks(-Vel(:,3),'MinPeakDistance',100);
[PKSL,LOCSL]= findpeaks(-Vel2(:,3),'MinPeakDistance',100);

for k=1:size(LOCSL)-1

TS.Lstridetime(k)=(LOCSL(k+1)-LOCSL(k))/100;
TS.Lstridelength(k)=Position2(LOCSL(k+1),1)-Position2(LOCSL(k),1);
TS.Lstridespeed(k)=TS.Lstridelength(k)/TS.Lstridetime(k);

end
for k=1:size(LOCSR)-1
TS.Rstridetime(k)=(LOCSR(k+1)-LOCSR(k))/100;
TS.Rstridelength(k)=Position(LOCSR(k+1),1)-Position(LOCSR(k),1);
TS.Rstridespeed(k)=TS.Rstridelength(k)/TS.Rstridetime(k);
end
for k=1:size(LOCSR)
    for z=1:size(LOCSL)-1
if LOCSR(k)> LOCSL(z) && LOCSR(k)< LOCSL(z+1)
TS.Lsteptime(k)=(LOCSL(z+1)-LOCSR(k))/100;
TS.Rsteptime(k)=(-LOCSL(z)+LOCSR(k))/100;
TS.Lsteplength(k)=Position2(LOCSL(z+1),1)-Position(LOCSR(k),1);
TS.Rsteplength(k)=-Position2(LOCSL(z),1)+Position(LOCSR(k),1);
TS.Lstepspeed(k)=TS.Lsteplength(k)/TS.Lsteptime(k);
TS.Rstepspeed(k)=TS.Rsteplength(k)/TS.Rsteptime(k);
end
    end
end

[PKSR2,LOCSR2]= findpeaks(Vel(:,3),'MinPeakDistance',100);
[PKSL2,LOCSL2]= findpeaks(Vel2(:,3),'MinPeakDistance',100);
for k=2:size(LOCSL2)
  for z=1:100
   if Vel(LOCSL2(k)-z,3)-Vel(LOCSL2(k)-z-1,3)<0
   break
   end
TS.Lskatetime(k-1)=(LOCSL(k)-z-LOCSL(k-1))/100;
  end
end

for k=2:size(LOCSR2)-1
  for z=1:100
   if Vel(LOCSR2(k)-z,3)-Vel(LOCSR2(k)-z-1,3)<0
   break
   end
TS.Rskatetime(k-1)=(LOCSR(k)-z-LOCSR(k-1))/100;
  end
end
Lstridetime=TS.Lstridetime;
Lstridelength=TS.Lstridelength;
Rstridetime=TS.Rstridetime;
Rstridelength=TS.Rstridelength;
Lsteptime=TS.Lsteptime;
Lsteplength=TS.Lsteplength;
Rsteptime=TS.Rsteptime;
```

```
Rsteplength=TS.Rsteplength;
Rspeed=TS.Rstridespeed;
Lspeed=TS.Lstridespeed;
end
```

G5. Marker Plots

```
clc
clear all
close all
[A.num,A.txt,A.raw]=xlsread('Trial1.xlsx');
Mtrajectories = A.num(5:end,:);
ylabeling={'x [mm]','y [mm]','z [mm]'};
Mtrajectories(:,1)=Mtrajectories(:,1)/100;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('units','normalized','outerposition',[0 0 1 1])
sgtitle ('Right Foot');
for i=1:4
for j=1:3
    subplot(3,1,j)
    plot(Mtrajectories(:,1),Mtrajectories(:,3*i+j-1)); hold on;
    xlabel('Time [s]');
    ylabel(ylabeling(j));

end

end
hold off;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('units','normalized','outerposition',[0 0 1 1])
sgtitle ('Right Shank');
for i=5:9
for j=1:3
    subplot(3,1,j)
    plot(Mtrajectories(:,1),Mtrajectories(:,3*i+j-1)); hold on;
    xlabel('Time [s]');
    ylabel(ylabeling(j));
end
end
hold off;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('units','normalized','outerposition',[0 0 1 1])
sgtitle ('Left Shank');
for i=10:14
for j=1:3
    subplot(3,1,j)
    plot(Mtrajectories(:,1),Mtrajectories(:,3*i+j-1)); hold on;
    xlabel('Time [s]');
    ylabel(ylabeling(j));
end
end
hold off;
figure('units','normalized','outerposition',[0 0 1 1])
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sgtitle ('Left Foot');
```

```
49 for i=15:18
50 for j=1:3
51
52     subplot(3,1,j)
53     plot(Mtrajectories(:,1),Mtrajectories(:,3*i+j-1)); hold on;
54     xlabel('Time [s]');
55     ylabel(ylabeling(j));
56 end
57 end
58 hold off;
```