Quantifying Upper-Limb Bimanual Coordination Performance Using Machine Learning Techniques for Concussion Screening

by

Dalya Bassam Al-Mfarej

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2021

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Current concussion screening and diagnosis tools rely on symptom checklist scores, along with subjective assessments performed by a clinician. This introduces high variability and bias, and increases the chances of missed diagnoses, which could lead to inappropriate return to play decisions resulting in dire consequences on the athlete's health, especially in the case of a second hit. While limited, objective measures for motor assessment exist, they are generally infeasible for use as sideline screening tools.

Impaired bimanual motor coordination is one of the major motor deficits that individuals with a concussion can experience. Measuring the degree of impairment in bimanual coordination could be an effective metric for concussion screening. These metrics would provide objective means for sideline screening that are more effective than the currently employed subjective assessments. However, bimanual coordination metrics that are sensitive to concussion remain unknown. Furthermore, a feasible testing paradigm that permits obtaining such metrics on the sideline is also lacking. This thesis contributes the design and evaluation of a novel tool that can be implemented and used in practice, on the sideline of sporting events, to detect coordination impairment associated with concussion objectively.

In the first component of this work, a novel testing paradigm for bimanual motor control assessment is proposed and analyzed. Comprising of a simple 1Hz in-phase vertical bimanual movement, the proposed bimanual coordination paradigm requires individuals to integrate multiple sources of sensory information simultaneously (i.e., visual, auditory) to produce a successful arm coordination pattern. The most informative metrics or features, including power-based features, and average peak-to-peak distance were extracted and analyzed to identify metrics that are sensitive to motor deficits, pointing to potential concussion. A machine learning model was developed to distinguish athletes with a concussion and on-going symptoms (CON-S) from healthy controls (HC) using the extracted features from their kinematic data. The proposed method was able to identify concussion with an average accuracy of 86% using a logistic regression model, and 88% using an Adaboost classifier.

Issues arise with difficulties in acquiring the required kinematic data, wherein tools currently in use for such applications are expensive, limited to laboratory settings and time consuming. Current gold standard methods are dominated by motion capture, which significantly limits the feasibility of using the proposed paradigm on the sidelines. As such, a portable, cost-effective, and rapid method for data collection is essential. One promising alternative is the utility of computer vision techniques. Utilizing such a method would allow data collection to be performed using devices with a camera, such as a smart phone, in a wide range of settings or environments, without the need for extensive calibration or markers like a motion capture system. In the second component of this thesis, a collection tool utilizing computer vision is proposed and tested for kinematic assessment, and its accuracy was compared to a research-grade motion capture device. Using a video sampling at 120 fps, an average peak-to-peak error of 6.96 mm was obtained.

The overall proposed system utilizes computer vision to measure arm motion kinematics and assess bimanual motor coordination, which is expected to deteriorate following a concussion. Proof of concept analyses indicate that the extracted features are able to identify concussion effectively. This tool would be of benefit for quick, portable, and objective sideline concussion screening which has the potential to reduce missed diagnoses and inappropriate return to play decisions to prevent further injury.

Acknowledgment

I would like to thank my amazing supervisor Dr. James Tung for his continuous guidance throughout my studies. I would also like to extend my gratitude to my lab members for their help and to my family and friends for their constant support and encouragement.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Concussion is defined as “a complex pathophysiological process affecting the brain induced by traumatic biomechanical forces.” [1]. As a type of traumatic brain injury (TBI), concussions are considered as mild to moderate on the spectrum of injury severity [2]. Occurring from direct blows to the head, blasts, or anoxic injuries [3,4,5], sport-related concussions make up approximately 20% of all head injuries reported in the United States alone [6]. Annually, 1.6-1.8 million individuals are estimated to sustain sports-related concussions in the United States [7], making concussions a highly common public health problem, particularly in youth athletes [7]. Due to reliance on subjective measures and self-reported symptoms in sports-related concussion screening, about 30-50% of concussions go unreported [8,9]. This increases the chance of the athlete sustaining a second hit, resulting in more dire consequences (i.e., second impact syndrome). This highlights the lack of sensitive measures for detecting a potential concussion, specifically on the sidelines of sporting events. A study performed with 97 athletes showed that individuals who sustained a concussion but were not immediately removed from the game were 2.2 times more likely to have a lengthier recovery [8]. This is supported by other research, emphasizing the need for timely removal following a suspected concussion [9, 10].

Following an initial injury, the brain is exponentially more vulnerable to damage, which makes sustaining a second injury significantly more dangerous [8]. This window of vulnerability wherein physical exertion and/or subsequent impacts following an initial concussion could result in dire consequences that will negatively impact recovery, also known as second impact syndrome [8]. Due to this risk, it is essential that a suspected concussion is recognized rapidly, and that the athlete is appropriately screened for concussion, before being allowed to return to the game. This rapid screening is typically conducted on the sidelines, often relying on an athletes’ self-reported symptoms. Self-reported symptoms pose an issue, as athletes have been known to not disclose their symptoms, or do so too late, to avoid being removed from the game [8]. Moreover, often the athletes do not recognize their symptoms or believe they are not serious enough to be disclosed [8, 11, 12]. Relying purely on symptom reports does not account for other complex dysfunction(s) that could have resulted from the concussion but may not be evidently displayed as observable symptoms. Considering the critical role of sideline screening and the limitation of self-report measures reinforces the need for sensitive screening and diagnostic measures able to detect functional deficits objectively.

Diagnosing a concussion on the sideline of a sporting activity is a very challenging task, not only due to the brief evaluation time and reliance on subjective observation of the athlete and their self-reported symptoms, but also due to the pressure to return the athlete to the field [11]. Also, athletes may not exhibit and/or have been known to mask obvious signs of concussion [11]. Moreover, many levels of sports do not have a dedicated medical personnel for concussion screening on the sideline, which requires the athletes and coaches to recognize and screen for potential concussion [12].

Currently, symptom checklists are the most commonly used method for concussion screening on the sideline, which is usually paired with observational assessment. These assessments revolve around assessing the athlete's cognitive or/and motor abilities [13], while observed by an assessor (e.g., counting errors while performing balance test), which results in high bias and variability among the different assessors. Based on the symptom checklist and observational assessments, if the athlete is believed to have a concussion, they are then examined by a medical professional. In addition, when the athlete is asymptomatic and able to perform cognitive testing, neurocognitive batteries are used to assess recovery, such as ImPACT (Immediate Post-Concussion Assessment and Cognitive Testing), a computerized tool for assessment of cognitive function [14, 2]. Another well-utilized tool is the SCAT-5, a standardized concussion assessment tool encompassing several motor and cognitive assessments, all of which are conducted on observation [15]. SCAT-5 assessments include symptom checklist, memory test, and a BESS (Balance Error Scoring System) assessment, among others. One of the motor skills assessed is bimanual coordination, which often deteriorates following concussion. Coordination is assessed using the SCAT-5 tool by performing the finger-to-nose (FTN) coordination test, along with observing signs of motor incoordination [15]. The FTN test requires the participants to first extend their arms fully, touch the tip of their nose, then return finger to extended position, successively and as fast as they can. Figure 1 below illustrates how the task is performed [16]. The athlete then receives a score of 1 (i.e., passed the coordination test) if they are able to perform 5 repetitions of this task correctly in 4 seconds or less. If they are unable to complete this number of repetitions in the given time, or if they do so incorrectly (i.e., do not touch their nose or do not extend their arms fully) they are given a score of 0 which indicates they failed the coordination test [16, 17]. The FTN uses counts of errors, or repetitions in a given time to produce confident yes/no decisions on coordination impairment. While the FTN is rapid and easy to administer, the major drawback from this test is its poor sensitivity, specifically for detecting mild motor impairment that is not visible/observable. Moreover, it relies on observation, subjective judgment, and decision could vary depending on the assessor. Lack of test sensitivity increases the chances of missed diagnoses and inappropriate return to play decisions.

To address the drawbacks of this method, a study proposed the use of an instrumented FTN test for assessment of Cerebellar Ataxia [18]. A sensor was placed on the finger and hand movements were measured, while performing the FTN procedure. The study collected measurements for 9 instrumented tasks and used the combined data for objective assessment. Features such as the resonance frequency and amplitude were then extracted from the FTN task and were used with features from the other instrumented tasks to build a classifier, yielding an accuracy of 97%. The study also included a clinician's assessment for the same tasks and noted some discrepancies in motor feature importance that were selected by the model and the clinician. The instrumented FTN method has not been studied for concussion assessment, therefore its performance on detecting a concussion is unknown. Furthermore, the model used in the discussed work utilized features from multiple motor assessment tasks used conjointly for classification. As such, the performance of the FTN test alone on model accuracy is lower than the reported accuracy of the combined model.

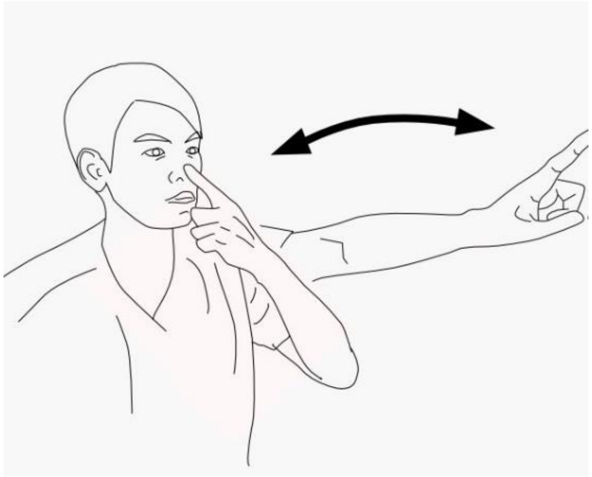


Figure 1: Finger-to-nose test currently implemented for detecting impairment in bimanual coordination [19]

Acute symptoms that are commonly demonstrated by individuals with a concussion include headache, confusion, loss of balance, memory impairment, incoordination, disturbances in vision, and changes in reaction time, behavior, and cognitive abilities [12]. This thesis is focused on detecting impairment in bimanual motor coordination following a concussion. The literature discussed hereafter will primarily focus on studies that have been conducted for assessment of bimanual motor coordination specifically with regards to concussion. Note that these methods have not yet been implemented in practice for concussion screening or diagnosis.

In one such study, assessment of bimanual coordination was conducted by performing an object hit task, using a robot, which was used to measure upper limb extremity performance [19]. Postural stability was also assessed conjointly using a force plate. The research study incorporated bimanual motor control of upper limbs in conjunction with visuospatial attention to study motor impairment present in individuals with a concussion. A robotic arm (KINARM) was utilized, and a task that is commonly performed for assessing stroke was performed in this assessment. The participants were instructed to grasp the robot handles and to ‘‘use both hands to hit as many [virtual] objects away from the bottom of the screen as possible”. The study concluded that there were differences observed between the concussed and the healthy groups, wherein objective measures were obtained and shown to have potential in quantifying upper extremity motor performance deficits in a subset of athletes with concussion. The main features derived for assessment were the total number of hits and the hand speed [19]. A limitation of this study is that it is not feasible for use on the sideline of sporting activities as the method requires expensive and specialized equipment, is not portable, and requires trained personnel for administration.

In another study, a computerized measure utilized the Automated Neuropsychological Assessment Metrics (ANAM), which involved finger tapping to measure reaction time and tapping frequency of each hand was implemented to assess bimanual motor coordination [20]. This study measured the bimanual two-choice reaction time in concussed and healthy participants and found a trend indicating a measurable difference between the two groups. The presence of visual and kinesthetic sensory inputs has been shown to influence performance of bimanual coordination tasks and learning of the tasks [21]. Another study explored assessing motor skills using a bimanual coordination task to detect cognitive-motor deficits caused by concussion [21]. The paper utilized a joystick and a computer, and the participants were required to replicate coordination patterns that were presented to them on a computer screen, in the presence of several constraints. The results of the study suggest that bimanual coordination deficits are present in individuals with concussion, which can be objectively measured using kinematics.

In-phase bimanual coordination tasks (where the limbs are moved synchronously) are more likely to be used for concussion screening, as they produce distinguishable motions due to their stricter requirements for spatial and temporal coupling of the hands [21]. Detected deficits in the bimanual control skills are likely attributed to poor spatiotemporal coupling in eye-hand coordination tasks, following a concussion [21]. Finally, the study author noted the important role that an auditory metronome plays on the performance of a bimanual coordination task [21]. The study reported the difference in performance induced by varying the frequency of movement, such that a higher frequency reduces stability of the motion.

1.2 Thesis Objectives and Outline

The main goal of this thesis is the design and evaluation of a tool that can be implemented and feasibly used in practice, on the sideline of sporting events, to detect concussion.

Drawing on findings from previous research studies, a novel and rapid testing paradigm was implemented for assessing upper limb bimanual motor coordination to classify concussion. The designed paradigm incorporates visual and auditory stimuli (spatiotemporal) to guide the motion of the upper limbs, using a simple task.

The first objective of this thesis (Chapter 2) is to analyze a novel bimanual arm control testing paradigm to evaluate motor actions associated with a concussion that may be too subtle to assess via self-report or observation. More sensitive paradigms, such as the proposed bimanual coordination method, are needed to better inform objective diagnoses on sporting event sidelines. This novel task has the potential to alleviate inappropriate return to play decisions by providing the clinician or coach data-driven outcomes regarding an athlete’s motor impairment using a simple and rapid motor assessment procedure. The goal is to use this tool for initial screening in case of a suspected injury, to inform decisions as to whether the athlete is safe to return to play or if they require further medical attention and should be taken out of the game. Furthermore, this tool could be used as a habitual screening tool wherein athletes are assessed or monitored regardless of a suspected injury. The integration of this tool with other objective measures, such as a modified BESS using measurable quantitative data or computerized cognitive assessment tools, would yield better predictions for better informed diagnoses.

The second portion of this thesis (Chapter 3) aims to develop and test a novel data collection system that will allow for feasible implementation of the proposed bimanual coordination paradigm, on the sideline of sporting events. This novel system involves utilizing computer vision to enable a rapid, low-cost, and easy-to-use method for collecting limb kinematic data, using a smartphone camera.

The rest of the thesis is structured as follows:

- **Chapter 2** assesses the proposed bimanual coordination testing paradigm, wherein promising informative features are extracted and analyzed. In addition, the potential of building a classifier using the bimanual control data, for identifying motor impairment due to concussion, is examined.
- **Chapter 3** describes the development and testing of a computer vision-based system for kinematic assessment of upper limb movements to permit concussion screening on the sidelines of sporting events.
- **Chapter 4** summarizes the work conducted in this research, states current limitations and potential improvements for future work

2 Objective Identification of Motor Impairment Following Concussion

2.1 Introduction

Current concussion diagnostic protocols rely heavily on subjective measures, particularly on the individual's ability to communicate symptoms with the healthcare personnel. This reliance on subjective self-report results in inconsistencies and missed diagnoses which could be detrimental to the athlete's health. There are very few effective sideline concussion diagnostic tools available, which are often limited to laboratory settings due to equipment operating condition requirements making them ineffective for sideline screening. Such tools have been developed to measure balance, eye-tracking for vestibulo-ocular screening, reaction time and/or visuomotor speed [22]. Other tools such as helmet mounted head impact sensors, and blood biomarkers are also active research areas [22]. However, such tools are difficult to implement and/or are invasive and time consuming for practical use on the sidelines. In terms of the motor assessments, balance is the most commonly studied, wherein several have proposed and shown effective the use of wearable sensors for assessment of balance [22, 23, 24]. This generally involves performing a modified BESS (mBESS) wherein a sensor (IMU/accelerometer) is mounted on the athlete (typically near the sternum region or the lower back), and acceleration data is used to obtain descriptive features that are able to distinguish impaired balance associated with concussion. Another method for assessing postural stability was proposed wherein two Microsoft Kinect sensors were used to obtain human body skeleton positions, which were then tracked to detect balance errors, similar to the BESS assessment [25].

Eye-tracking paradigms have shown to be effective in detecting concussion within days after play, but not enough evidence has been presented to warrant their efficacy for use on the sideline for rapid assessment [22, 26]. Mobile device applications have been investigated to measure reaction time and visuomotor speed, but such tools have shown discrepancies when compared to traditional tools for similar assessment (ex., SCAT5 and the SCAT mobile app) [22]. For effective sideline concussion screening, the integration of multiple tools to measure several different aspects of impairment, is essential.

One of the major long-term motor skill impairments that has been understudied with regards to concussion screening compared to other motor impairments is bimanual coordination. Some research has been done showing the effectiveness of assessing sensor-based bimanual motor coordination for detecting concussion (as discussed in Chapter 1). However, feasible sensor-based methods for use on the sidelines have yet to be widely used. Current methods for bimanual motor coordination assessment on the sidelines involves some version of the finger-to-nose test, which relies on observation, with less sensitivity and reliability. Using previous findings demonstrating individuals with a history of concussion show motor performance deficits when attending to and integrating multiple sources of afferent information, this chapter investigates new features that measure deficits in performance associated with concussion.

The research questions posed in this chapter are: Can kinematic data from a simple upper body bimanual motor control task be employed to identify concussion? What features in the kinematic data distinguish athletes with and without a concussion? Can we generate a model to identify those patterns and detect impairment associated with concussion using this paradigm?

The hypothesis predicts that individuals who have a history of concussion would demonstrate objectively poorer performance in a bimanual coordination task compared to those who did not have a history of concussion, and that this difference could be distinguished using a classifier. Furthermore, it is hypothesized that, due to the simple nature of the motion, small contributions from several features will be required for accurate predictions. This will be assessed by building machine learning models for classifying individuals with and without a concussion, analyzing their performance, and the optimal number of features/metrics selected for classification.

The proposed upper body bimanual coordination assessment paradigm is differentiated by task demands requiring integration of multiple sensory systems in conjunction with on-line motor planning and execution to coordinate both limbs in space and time and produce a successful movement. Although these tasks are not traditionally thought of as complex, participants were required to process multiple sources of sensory information, including visual, auditory, and tactile/proprioceptive information, of limb locations in comparison to the targets, auditory information, and proprioceptive information to maintain a stable phase movement. Therefore, instead of dividing attention to delineate different tasks or movements, participants were required to attend to the stimuli and integrate sensory information to perform the desired coordinated movement. Such a screening method would rely on objective metrics that reflect motor deficiencies often obtained following a head injury, rather than relying on subjective human perceived irregularities in motor skills.

2.2 Data Collection

The data used in this chapter was obtained from two separate data collection sessions, following the same collection protocol. The data was collected by collaborators Dr. Dave Gonzalez and Alex Scherger from University of Waterloo Kinesiology department [27].

Data was obtained from 48 varsity level athletes. In total, 19 athletes had no history of concussion (HC), 18 athletes had a concussion and were currently symptomatic (CON-S), and 11 athletes had previous concussion but were asymptomatic (CON-A). All concussions were diagnosed by a sports medicine doctor. Following concussion diagnoses, athletes completed the ImPACT post-concussion testing. Prior to assigning a participant to the concussion group, they were required to complete an interview wherein they disclosed further information regarding the physician diagnosis, how they obtained the injury, and ongoing symptoms. The participants were required to complete two questionnaires documenting their concussion history on two different dates, to ensure consistency of the self-reported symptoms. The participants were also asked to complete an additional 22 question symptom checklist, in accordance with previous published research studies [28,29]. An Office of Research Ethics committee at the University of Waterloo approved the study.

2.2.1 Apparatus

The participants were instructed to sit on an adjustable chair facing towards a desk, as shown in Figure 2. On the desk, a visual guide consisting of a vertical rod with two protruding markers placed 180 mm apart was situated to the right of the participant's side (Fig. 2, A). The purpose of the visual guide is to constraint the participant's motion to the boundaries set by the protruding markers. Participants were instructed to maintain their motion within the boundaries; however, they were instructed to not touch the boundary in order to avoid providing tactile feedback. For motion capture using Optotrak (NDI Digital, Waterloo, ON), 2 infrared emitting diodes were placed on the tip of the participant's index fingers (distal phalange), on both hands (Fig. 2, C). In addition, the use of an auditory guide was also incorporated, wherein the participants were required to synchronize their motion to a 1 Hz monotone beep. As such, visual and auditory sensory feedback were provided to guide the motion. E-prime software 1.1, Optotrak 3D Investigator (NDI digital, Waterloo, ON) recording at 200 Hz was used.

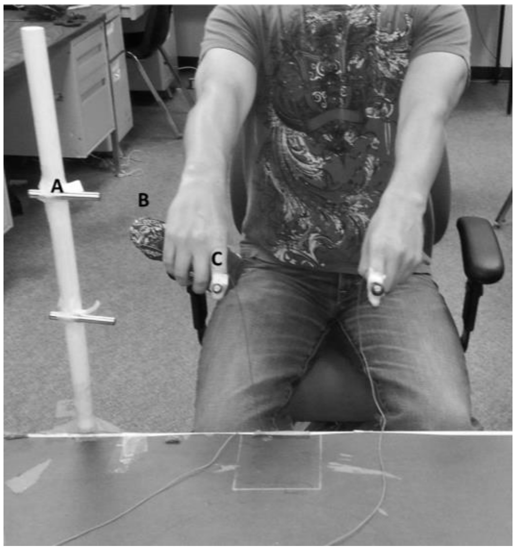


Figure 2: Data collection apparatus. Label A shows the visual guide and C denotes the placement of the marker.

2.2.2 Procedure

The participants were instructed to fully extend their arms all the way to the index finger and close the remaining fingers in a loose fist. The participants were then asked to perform a simple vertical arm movement, wherein they were required to move both their arms up and down at the same time (in phase), within the visual boundary, and at a frequency of 1 Hz. The movements were performed such that the downward peak corresponded to the timing of the metronome beep. Each trial lasted about 10-12 seconds, and four trials were collected per participant.

2.3 Extraction and Analysis of Features for Identifying Motor Impairment

In this section, methods to extract features to distinguish the concussion groups (CON-S & CON-A) from the healthy controls (HC) are described. These features will be analyzed for potential relationship to the classification problem (CON-S vs HC) using feature analysis methods. The extracted features will then be employed to generate and test classification models.

2.3.1 Feature Extraction

For each of the 4 trials collected from every participant, a set of features were obtained. Position data collected for each participant were truncated such that all the signals obtained were the same length, typically including about 8 cycles per trial. Missing datapoints were filled using shape-preserving piecewise cubic spline ('pchip' function in MATLAB) interpolation, and the mean (offset) was subtracted. The data was then filtered using a 3rd order Butterworth filter with a cut-off frequency of 15 Hz.

Due to the lack of prior information about the movement and its potential relationship to concussion, extensive feature set extraction was conducted to gain insight into its behavior. Following filtering, 24 time-domain and 33 frequency-domain features were computed for each hand. The median of the four trials was computed to obtain a single feature array for each hand, for every participant. The median was chosen to reduce the effect of outliers or 'bad' trials and obtain the average performance for each individual. To eliminate handedness dependency, and to avoid any uncertainties in labeling during data collection, the mean of the two hands for each feature was computed. In addition, the absolute value of the difference between the right and the left hand was also obtained. The combination of both the mean and the difference between the right and left hand features for each participant was then used to yield 85 total features (37 time-domain and 48 frequency-domain). Note that 19 of the features (8 frequency domain, and 11 time domain) were readily computed using both the right and left hand signals, and no averaging or differencing between the two hands was required. MATLAB (R2018a) was used for all signal processing tasks in this chapter. Some of the obtained features are correlated; however, that will be taken into account when selecting the best set of features. In the subsequent section, the feature extraction process is expanded on to illustrate the signal descriptors obtained.

The trajectory signal of interest is a power signal, that is windowed/truncated in time. It is a real-world signal, representing human motor control, which is a stochastic process. The acquired/measured signals cannot be predicted a priori. Due to the nature of the collection paradigm, it is assumed that the signal resembles a sine wave with noise or background components, where the amplitude, and phase of the sine wave (harmonic component) are independent random variables. The phase is a uniformly distributed random variable in the range 0 to 2 π.

In this thesis, the bimanual task process is assumed to be ergodic. The process is also wide-sense stationary as the first and second order statistics are time-invariant. The validity of these assumptions is required for the correlation functions and the power spectral density (PSD) estimate.

2.3.1.1 Time Domain Features

Thirty-seven (37) time-domain features were extracted from the position data for each hand. Time-domain features rely on the time-series signal for extracting descriptors and are generally concerned with changes pertaining to the signal's amplitude with respect to time. The features obtained are described in detail in Table 1 below.

Autocorrelation and Cross-correlation

The autocorrelation function measures the correlation between a signal and delayed copies of itself. In other words, it measures correlation at different lag or delay values of the random signal. In this work, autocorrelation is normalized by the length of the time-series. The root mean square (RMS) of the autocorrelation, and the highest 4 peaks and their locations (except for the first peak as it is equivalent to the total power of the signal, which is obtained in the next section, and always occurs at lag 0), were extracted as features.

The cross-correlation is similar to autocorrelation, but two different signals are compared rather than one signal to itself. Cross-correlation measures the correlation between one signal and delayed copies of another signal across a range of delays/lags. Cross-correlation is also normalized by the signal's length. This was used to compare the left and right hand signals for each trial. Similarly, The RMS and the highest 4 peaks and their locations were obtained as features. Note that the location of the highest peak is also equivalent to the time delay between the signals.

Table 1: Time-domain features extracted from the signal

Measure	**Feature**	**Method**
Peak-to-Peak distance	- Mean - Standard deviation	$\mu = \frac{1}{N}\sum_{i=1}^{N} peak(i) - valley(i)$ $std = \sqrt{\frac{\sum_{i=1}^{N}[(peak(i) - valley(i)) - \mu]^2}{N}}$
Peak velocity	- Mean - Standard Deviation	$\mu = \frac{1}{N}\sum_{i=1}^{N} peak\_v(i)$ $std = \sqrt{\frac{\sum_{i=1}^{N}[peak\_v(i) - \mu]^2}{N}}$ where peak includes -valleys
Autocorrelation	- RMS (rms_{ff}) - Highest four peaks and their location (excluding the first peak)	$R_{ff}(k) = \begin{cases} \frac{1}{N}\sum_{n=1}^{N-k} f_n f_{n+k}\,, & k \geq 0 \\ R_{ff}(-k), & k < 0 \end{cases}$ where f is the signal of interest, N is the length of the signal and k is the lag $rms_{ff} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} Rff_n^2}$
Cross-correlation	- RMS (rms_{fg}) - Highest five peaks and their location (location of the first peak is the delay between the two signals)	$R_{fg}(k) = \begin{cases} \frac{1}{N}\sum_{n=1}^{N-k} f_n g_{n+k}\,, & k \geq 0 \\ R_{fg}(-k), & k < 0 \end{cases}$ where g and f are the right and left hand signals, N is their length, and k is the lag. $rms_{fg} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} Rfg_n^2}$

2.3.1.2 Frequency Domain Features

Frequency domain signal representation was also of interest for feature extraction. Due to the nature of the signal, a high power spectral component is expected to be present at the 1Hz frequency bin as participants were instructed to synchronize their motion to a 1Hz monotone beep. However, it was hypothesized that there will be variations in this frequency, and potentially a pattern for distinguishing the two groups (HC vs CON-S). Forty-eight (48) frequency-domain features were extracted and fell into 5 general categories: Power spectral density (PSD), background spectrum, spectral kurtosis & instantaneous frequency, coherence & phase difference, and spectral entropy. The PSD and background spectrum provide information about the power distribution of different frequency components in the signal and were computed using a method discussed below. Kurtosis and instantaneous frequency account for changes in frequency content of the signal over time. Finally, coherence and phase difference provide a comparison between the right and left hand signals in their frequency domain representation. The frequency-domain features obtained are described in detail in Table 2 below.

Power spectrum

Given one realization of the stochastic process, PSD can be estimated to analyze the power distribution with respect to specific frequency bands. All PSD methods mentioned in this work are non-parametric. There are several methods for estimating the PSD, each with different resolution-variance trade-offs. Frequency resolution pertains to the ability to distinguish frequency components that are close to each other. Low frequency resolution leads to smearing of peaks in close proximity with each other, which leads to high bias.

Note the signals were truncated to have the same length for all participants to produce the same frequency resolution for all, as the sampling rate was consistent. The frequency resolution depends on the length and type of window used. A longer window allows for smaller mainlobe width thereby enabling finer frequency resolution and reducing spectral smearing. The grid resolution or the bin width of the DFT is the sampling rate divided by the length of the discrete Fourier transform (DFT). The length of the DFT is not necessarily the same as the signal length as it is often zero-padded to produce smoother looking peaks. However, the frequency resolution cannot be improved by zero-padding the signal, padding merely interpolates the frequency data for ease of interpretability. The zero-padding factor was also kept consistent for all participants in an effort to remain consistent among trials. Note that the grid resolution represents the minimum resolution of the DFT, however this needs to be multiplied by a coefficient that represents the increase in bandwidth due to the window (loss in resolution), for determining the ability to distinguish spectral components [30].

The simplest type of PSD is the periodogram wherein the signal is multiplied by the default rectangular window and the squared magnitude of the DFT is obtained. This method yields very high variance and spectral leakage. Note that spectral leakage and low frequency resolution both contribute to bias in PSD estimate. A modified version is the modified periodogram wherein another window is multiplied by the data to reduce spectral leakage; however, this still does not solve the variance issue. Windows pose a restriction on the achievable frequency resolution, based on their mainlobe width, which is often a trade-off with spectral leakage.

The periodogram is an inconsistent estimator as the variance does not decrease with increasing signal length. This issue is addressed by averaging over time ensembles of the signal to reduce variance in the PSD estimate and produce an estimator whose bias and variance decrease. Called Welch's PSD estimate, the signal is divided into n number of segments, and each segment is then multiplied by a window/taper, followed by computing the DFT. The results are magnitude squared to obtain a PSD estimate for each segment, and finally the PSD estimates of the various segments are averaged. In addition, overlapping windows are employed to allow for more segments to average and to reduce data loss at the window edges. Since different parts of the signal are used to compute the PSD for each segment, the results are approximately uncorrelated. Welch's method was used in this work for estimating coherence between the right and left hand signals. The general equation for Welch's PSD estimation is shown below:

$$\hat{S}_x^W(\mathrm{w}_k) = \frac{1}{K}\sum_{m=0}^{K-1}\frac{\Delta t}{N}\left|FFT_{N,k}(x_m)\right|^2$$

where K is the total number of segments, k is the segment number, N is the length of the window/segment, m is the window/segment number and Δt is the sampling interval. One issue with Welch's estimate is the loss in resolution due to shorter signal segments used for estimation, rather than the entire signal. The longer the signal/time-series, the higher the frequency resolution is. To address the issue, the Multitaper PSD estimate (MT), uses various PSD estimates of the entire signal then averages them.

Multitaper PSD Estimate

MT PSD estimate obtains various unique PSD estimates of the signal by using a special class of orthogonal windows/tapers known as the Slepian sequences [31, 32, 33]. The Slepian sequences (also known as the Discrete Prolate Spheroidal (DPSS) sequences) have the special properties: orthogonality, and optimal time-bandwidth concentration. These properties allow for the computation of various approximately uncorrelated PSD estimates of the signals, by multiplying the signal with a set number of Slepian tapers, then obtaining the PSD estimate for each, and finally averaging the unique estimates. This allows for the use of the entire signal, rather than segments of the signal like in Welch's method, thereby allowing for a higher frequency resolution, while also reducing the variance by utilizing averages. Another advantage of this method is that the Slepian tapers cover more parts of the signal thereby accounting for signal edges, which are generally attenuated by other common windows/tapers.

The DPSS are optimal for concentrating the energy in a given bandwidth [-W, W] for a finite signal/time-series. Each sequence (eigenvector) has a corresponding eigenvalue which is a measure of the fraction of energy concentrated within the frequency interval -W and W. The first 2TW eigenvalues are nearly 1, then decay exponentially to 0. As such, given a signal length N and desired bandwidth resolution [-W, W], approximately 2TW-1 DPSS Slepian sequences/tapers will have energy concentration close to 1 (optimal). As such, the time-half bandwidth product, TW, is an important parameter to select for the MT estimation and will act to control the resolution-variance trade-off, where a higher TW value will decrease variance and resolution. Figure 3 below shows the first 4 Slepian sequences/tapers. The TW parameter should be selected based on the frequency resolution requirements for the problem of interest. For this work, an TW product of 2.5 was selected, such that $2 * 2.5 - 1 = 4$ Slepian tapers were used (tapers shown from k=0 to k=3 in figure 3 below) to obtain 4 PSD estimates, which were then averaged.

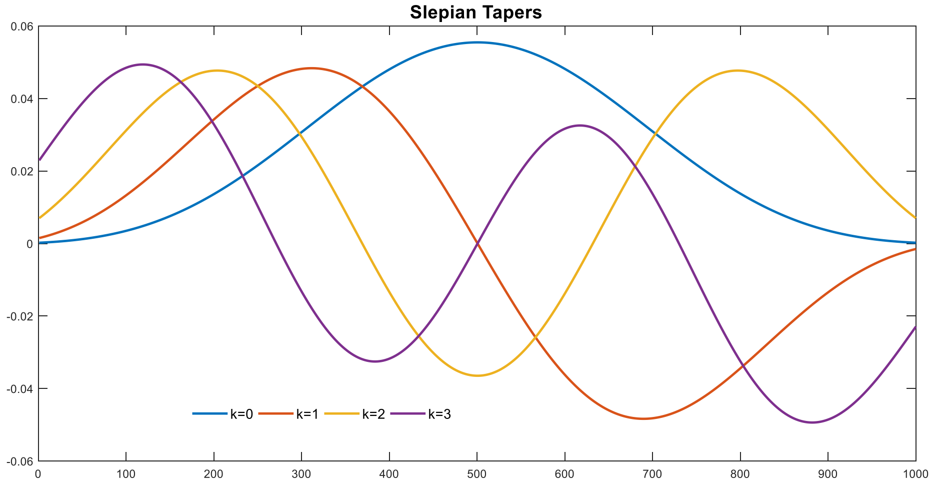


Figure 3: Slepian data tapers used for estimating the PSD using the MT method

The various PSD estimates are then averaged using an adaptive averaging method wherein the energy concentration of the Slepian sequence used to compute the specific PSD as well as the power distribution over the frequencies of the process are taken into account. The MT method was used to estimate the PSD in this work and obtain majority of the frequency-domain features.

The MT PSD estimate is solved iteratively using the equation shown below with the average of the two lowest order estimates (from the first two Slepian tapers) as a starting point:

$$\sum_{k=0}^{K-1} \frac{\lambda_k(\hat{S}_x^M(f)-\hat{S}_k(f))}{[\lambda_k(\hat{S}_x^M(f)+\hat{B}_k(f)]^2} = 0$$

where λ_k is the energy concentration (eigenvalue), $\hat{S}_k$is the PSD estimate for each taper-signal pair, $\hat{B}_k(f)$ is the broad-band bias, and $\hat{S}_x^M$is the final PSD estimate (after averaging). To find the frequency resolution of the estimated MT PSD, the following equation was used:

$$f_{bw} = \frac{TW}{N\Delta t}$$

$$f_{bw} = \frac{2.5 * 200}{1700} = 0.294\ Hz$$

$$Res = 2 * f_{bw} = 0.588\ Hz$$

where N is the length of the signal and Res is the frequency resolution [31]. The resolution obtained here is 0.588 Hz, where frequency components that are >0.588 Hz away will be distinguished from each other. As can be seen from the equation, a longer signal will lead to higher resolution. In this work, the band power, location and height of the highest peak, bandwidth, median frequency, and the lower and upper bounds of the bandwidth were extracted as features.

Reshaped spectrum

Another useful feature of the MT PSD estimate is the ability to extract line components (deterministic spectral components) from stochastic components and remove the line components to obtain the reshaped PSD. The reshaped PSD contains the stochastic background spectral components of the process. The F-test statistic is used to assess the presence of harmonic line components, at a probability level specified manually. The probability level used in this work is 99%. For full mathematical derivation of the MT method, refer to the papers published by Thomson and Chave [32,33].

In this work, the location of the first 2 line components, their power, as well as their F- test statistic were used as features. In addition, from the reshaped spectrum, the band power, median frequency, bandwidth as well as the first peak and its location were also extracted as features. The MATLAB function mwps was used for background spectrum estimation [34].

Instantaneous Frequency

To account for any potential non-stationarities in the signal and detect corresponding trends among the classes , the instantaneous frequency (IF) was obtained from the spectrogram . The spectrogram depicts how the frequency components of a signal change with time, similar to the power spectrum but for non-stationary signals whose frequency content is time-dependent. The instantaneous frequency is the average of the frequencies present in the signal for each time frame. For all the signals in this work, the slope of the instantaneous frequency line was small (all trials were <0.013 Hz/s), and the IF was around 1 Hz. For this work, the slope and the standard deviation of the instantaneous frequency were extracted and used as features.

Magnitude-squared Coherence

The coherence function computes the relationship between two signals as a function of frequency. A value closer or equal to 1 indicates high coherence between the two signals at that frequency, while a value closer or equal to 0 indicates low coherence between the two signals at that frequency. Coherence is computed using the PSD of each of the right and left hand signals and the cross-spectral density of the two signals. For the spectral densities computed for coherence, Welch's method was used. A Kaiser window with a beta value of 5, window length of L(signal)/3~566 samples, and 50% window overlap was used. As such, the frequency resolution obtained is ~ 1 Hz. The RMS, highest two peaks and their frequency location, and the minimum valley and its location were extracted and used as features.

Spectral Entropy

Spectral entropy is computed by finding the Shannon entropy of the signal's normalized power distribution in the frequency domain. The spectral entropy of the whole signal is extracted and used as a feature.

Spectral Kurtosis

Spectral kurtosis (SK) is used to identify transients in the frequency-domain signal. In other words, it is used to pinpoint non-stationarities in the signal, and the frequencies where they occur. SK is computed using the short-time Fourier transform (STFT), and a confidence level of 95% was used. For feature extraction, the frequency location of the highest kurtosis value above the threshold (computed from the 95% confidence interval), for frequencies lower than 10 Hz, was extracted and used.

Table 2: Frequency-domain features extracted from the signal

Measure	Feature	Method
PSD	- Highest three peaks and their location - Band power [0-10] Hz (BP) - Median frequency (MDF) - Occupied bandwidth (OB) - Upper and lower frequency bounds of the bandwidth - Harmonic and quasi harmonic spectral component's frequency, power, and F-statistic.	$BP = \sum_{f=0}^{10} \hat{S}_x^M(f)$ where $\hat{S}_x^M$ is the PSD estimate $OB = frequency\ band, 99\%\ of\ power$ $\sum_{i=1}^{MDF} P_i = \frac{1}{2}\sum_{i=1}^{M} P_i$ where MDF is the median frequency bin, P_i is the power at frequency bin i and M is the length of frequency bins. **For derivation of the spectral lines, background spectrum and the F-statistic, refer to [31,32]**
Background spectrum	- Band power [0-10] Hz (BP) - Median Frequency (MDF) - First peak and its location - Occupied bandwidth (OB)	
Kurtosis	- Frequency location where the kurtosis $K(f)$ crosses the 95% threshold, in the frequency range [0-10] Hz.	$K(f) = \frac{\langle \lvert S(t,f)\rvert^4 \rangle}{\langle \lvert S(t,f)^2\rvert^2 \rangle} - 2, \qquad f \neq 0$ where $S(t,f)$ is the STFT of the signal.
Instantaneous Frequency	- Slope and standard deviation of the instantaneous frequency $f_{inst}(t)$ from the spectrogram.	$f_{inst}(t) = \frac{\int_0^\infty f\ P(t,f)\ df}{\int_0^\infty\ P(t,f)\ df}$ where $P(t,f)$ is the spectrogram
Spectral Entropy	- Spectral entropy of whole signal	$SE = -\frac{\sum_{M=1}^{N} P(m)\log_2 P(m)}{\log_2 N}$ where P(m) is the probability distribution of the power spectrum and N is the total frequency points.

Table 3: Frequency-domain features extracted from the signal-continued

Measure	Feature	Method
Phase Difference	- Phase difference at the maximum frequency (fundamental) of each signal	$PD = \lvert \angle(DFT_1[f == 1]) - \angle(DFT_2[f == 1]) \rvert$
Coherence	- RMS (rms_C) - Highest two peaks and their location - Minimum valley and its location	$C_{xy}(f) = \frac{\lvert P_{xy}(f) \rvert^2}{P_{xx}(f) P_{yy}(f)}$ where $P_{xy}(f)$ is the cross power spectral density and $P_{xx}(f), P_{yy}(f)$ are the PSDs of each signal (right and left hand) $rms_C = \sqrt{\frac{1}{N} \sum_{n=1}^{N} {C_{xy}}_n^2}$

2.3.2 Feature Analysis

Following feature extraction, feature analysis was performed using statistical (filter/univariate methods) and machine learning methods (wrapper methods). The univariate methods used in this work are one-way ANOVA, Pearson correlation coefficient and mutual information. As for wrapper methods, the genetic algorithm was used. The purpose of feature analysis is to find the best subset of features for identifying a potential concussion. Note that the features selected in this section were not used in the subsequent section for building a machine learning model (different dimensionality reduction methods were implemented there) as the purpose of this section is to analyze the features. Python was used for all feature analysis.

Univariate Statistical Methods

Univariate feature selection methods reveal useful patterns between each feature independently, and the response variable (i.e., participant class/group). They do not account for feature interactions or the overall ability of multiple features to be used conjointly in predicting the response variable.

Firstly, highly correlated features with Pearson correlation of 0.99 or more were extracted and are summarized in the next paragraph. An ANOVA was implemented and all significant features (at a 5% significance level, such that their p-values were < 0.05) are listed below. To ensure ANOVA assumptions were not violated, normality of residuals and homogeneity of variance were visually assessed and were deemed acceptable for this analysis overall. For each of the top features selected by the ANOVA, the correlation coefficient is also listed. Note that the primary shortcoming of the univariate feature selection methods used here is their inability to account for interactions among features, as they only assess the importance of each feature independently, on the output. In addition, these methods only identify linear relationships with the response variable. Table 4 below shows the p-values and correlation coefficients for the top features selected. Note that none of the listed features are 100% correlated, as features that had a correlation coefficient of 1 with another feature were removed prior to ANOVA analysis.

Several of the power features from the PSD were highly correlated with each other (correlation of 0.99 or above). As such, any one of them is sufficient to include (i.e., band power, peaks in PSD, spectral lines). In addition, the cross-correlation peaks were highly correlated to the autocorrelation peaks. In turn, the autocorrelation peaks were also highly correlated with the power features from the PSD. However, the features from the reshaped spectrum did not show high correlation with the other features. The highly correlated features were still analyzed in this section as the aim is to analyze and understand kinematic descriptors of impairment. Highly correlated features will be filtered out when building a classifier in the subsequent section.

Table 4: Best features as per ANOVA and Pearson correlation coefficient analysis where the class/group is the dependent variable

Feature	**P-value**	**Pearson Correlation**
Average Band power of the Reshaped Spectrum	0.000767	0.529
Average standard deviation of the peak velocities	0.001051	0.517
Average bandwidth of the reshaped spectrum	0.004687	-0.455
Average value of F-test statistic for the first spectral line	0.008287	-0.428
Average mean of the peak velocities	0.010547	0.415
Peak 1 of cross-correlation	0.011156	0.412
Average Band power	0.013211	0.404
Average mean peak-peak distance	0.016915	0.390
Average max peak height of reshaped spectrum	0.017547	0.388
Peak 2 of cross-correlation	0.017800	0.388
Peak 3 of cross-correlation	0.017990	0.387
Average max peak height of power spectrum	0.018265	0.386
Average power of first spectral line	0.020137	0.381
Peak 4 of cross-correlation	0.020792	0.379
Peak 5 of cross-correlation	0.021352	0.377
Average Peak 2 of autocorrelation	0.023419	0.372
Average RMS of autocorrelation	0.025435	0.367
Difference in Band power of the Reshaped Spectrum	0.028264	0.361
Average Peak 4 of autocorrelation	0.030080	0.357
Difference in bandwidth	0.040619	-0.338
Average of standard deviation of the amplitude	0.049348	0.325
Difference in frequency value of the second spectral line	0.049349	-0.325

As a non-parametric alternative, the top features selected by mutual information (MI) are shown in Table 5 below. The mutual information method measures any statistical relationship between a given variable and the output, not necessarily a linear one. Non-parametric tests do not require the data to conform to strict statistical assumptions (generally less powerful than the parametric under some conditions). A higher MI score indicates a more informative feature.

Table 5: Features with the highest mutual information

Feature	**Mutual Information**
Average Band power of the Reshaped Spectrum	0.380
Average standard deviation of the peak velocities	0.226
Average bandwidth of the reshaped spectrum	0.217
Average value of F-test statistic for the first spectral line	0.126
Average lower-bound frequency of the bandwidth	0.215
Difference in band power of the reshaped spectrum	0.177
Average value of F-test statistic for the second spectral line	0.125
Difference in bandwidth of the reshaped spectrum	0.108
Cross correlation peak 1	0.088
Average mean peak velocity	0.085

Genetic Algorithm

Wrapper methods select the optimal set of features based on the performance of specific machine learning models with different subsets of the features. The genetic algorithm (GA) is type of evolutionary algorithm that mimics natural selection and survival of the fittest, generally to solve optimization problems. In this work, the genetic algorithm is used to select the best subset of features, based on a specified fitness function.

The genetic algorithm is stochastic in nature and takes time to converge to the desired outcome. Convergence is established by a pre-set criteria. In this work, the maximum number of generations after which the algorithm halts was set to 50. In addition, if the best individual is not changing for 10 generations, the optimization process will also be halted. Each individual is the underlying estimator, which was chosen to be a logistic regression model, with a different subset of features. The logistic regression model was cross-validated using 18 repeated stratified k-folds with 30 repetitions. 18 folds were selected as the positive class had 18 samples while the negative had 19 samples, in our dataset. As such, 18 folds would ensure that each fold has one of each class, except for one fold which has 2 from the positive class and one negative class. The regression model was not regularized.

The features are binary encoded such that they are set to 1 if they are included in the model and 0 if they are not. The individual features (the 1s and 0s) are the genes, with the set of genes for each individual defined as a chromosome. Initially, the genes in the chromosomes are randomly selected. A population is the collection of individuals, and the population number was set to 500. The population number remains the same across generations (iterations), but the individuals are different, and are selected based on their fitness. The fitness of an individual is a score of their performance and is used to eliminate and select certain individuals. Two different GAs were implemented in this work. The first was based on hard class assignments, as such accuracy was used as the fitness function. Accuracy was chosen mainly because it is intuitive, and because the dataset is balanced so there are no issues pertaining to class imbalance affecting the metric. The second is based on probabilistic predictions, and the Brier score was selected as the fitness function. Explained in a subsequent section, the Brier score ranges from 0 to 1, wherein a smaller score indicates better model probabilistic predictions.

The GA starts with the initial population, scores the individuals performances, then selects certain individuals for mating (random cross-over). The probability of cross-over was set to 0.8. Sometimes, a mutation is introduced, which is random change(s) in gene(s) of an individual to produce diversity. The probability of mutation was set to 0.05. Finally, fit individuals can survive to the next generation (elitist selection) to reduce the chance of losing good chromosomes. Note that there was no restriction set on the maximum number of features that could be selected. Figure 4 below shows the general workflow of the GA algorithm [35]. The genetic algorithm was implemented using the sklearn-genetic library [36]. Tables 6 and 7 show the results of the GA algorithm wherein the best subset of features selected, in three different runs with different random seeds is shown along with the fitness score of the selected model/individual. In the tables, A refers to the features obtain by averaging the right and left hand results, while D refers to features obtained by taking the difference of the right and left hand. Note that std stands for standard deviation. The reported scores are the mean cross-validation (CV) score for the selected individual. For reference, the mean of the CV score using the same classifier, but with the whole feature set is 48.79% and the average Brier score is 0.287.

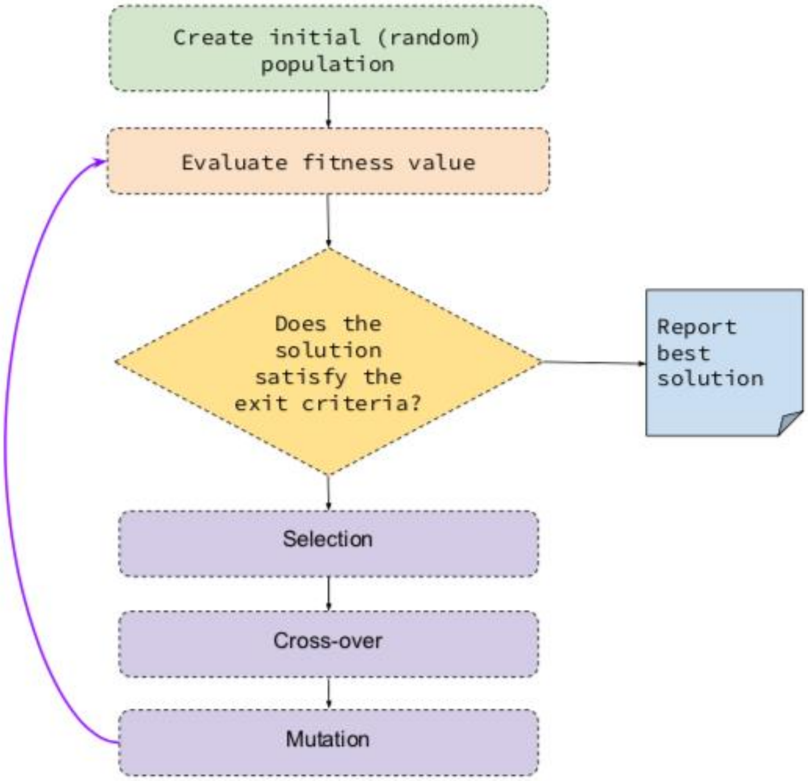

Create initial (random) population
Evaluate fitness value
Does the solution satisfy the exit criteria?
Report best solution
Selection
Cross-over
Mutation

Figure 4: General workflow of the genetic algorithm [35]

Table 6: Features selected by the genetic algorithm using the accuracy score as the fitness function. A-average, D-difference

# Selected	Score (Acc %)	Selected Features
13	79.0	A band power || A band power of reshaped spectrum || A bandwidth of reshaped spectrum || A peak 4 of autocorrelation function || D median frequency || D median frequency of reshaped spectrum || D peak 4 of autocorrelation function || coherence peak 1 || RMS of coherence function || delay || location of peak 3 in cross-correlation function || location of peak 4 in cross-correlation function || cross-correlation peak 5
15	75.9	D in std of peak velocities || A bandwidth of reshaped spectrum || A peak 3 of autocorrelation function || A location of peak 3 in autocorrelation function || A slope of instantaneous frequency || D frequency of first spectral line || D F-test statistic of line 1 || D bandwidth || D RMS autocorrelation || D peak of reshaped spectrum || D location of peak in reshaped spectrum || delay || cross-correlation peak 2 || location of peak 2 in cross-correlation function || cross-correlation peak 3
12	85.5	A band power || A of lower frequency of bandwidth || A bandwidth of reshaped spectrum || A peak 4 of autocorrelation function || A slope of instantaneous frequency || A std of instantaneous frequency || A peak height of reshaped spectrum || D in peak height of PSD || D in the lower frequency of bandwidth || D in std of instantaneous frequency || cross-correlation peak 4 || cross-correlation peak 5

Table 7: Features selected by the genetic algorithm using the Brier score as the fitness function. A-average, D-difference

# Selected	Score (Brier)	Selected Features
13	0.187	A band power of reshaped spectrum || A location of highest peak in PSD || A bandwidth of reshaped spectrum || A slope of instantaneous frequency || RMS of autocorrelation function || D location of highest peak in PSD || D location of peak 4 in autocorrelation function || D std of instantaneous frequency || D RMS of autocorrelation function || D location of peak in reshaped spectrum || location of the highest peak in coherence function || minimum coherence value || location of peak 5 in cross-correlation function
10	0.149	A band power of reshaped spectrum || A bandwidth of reshaped spectrum || location of peak 2 in cross-correlation function || A peak 4 of autocorrelation function || D median frequency of reshaped spectrum || D RMS of autocorrelation function || D location of peak in reshaped spectrum || peak 2 in coherence function || RMS of coherence function || delay
9	0.118	A band power of reshaped spectrum || A bandwidth of reshaped spectrum || A peak 3 of autocorrelation function || A RMS of autocorrelation function || D median frequency of reshaped spectrum || D bandwidth || D location of peak in reshaped spectrum || location of highest peak in coherence function || RMS of coherence function

2.3.3 Feature Analysis Discussion

Feature importance was analyzed using different methods, encompassing statistical (parametric and non-parametric), and wrapper methods. Using univariate statistical parametric and non-parametric methods, each feature is analyzed independently to determine its relationship with the response variable. As such, the interactions of features among each other and their collective influence on the response variable is ignored. Nevertheless, these methods provide a general overview of the predictors that have relevant patterns associated with the response variable.

Using parametric methods, namely p-value and correlation coefficients, the top 22 features were obtained. For all the listed features, their statistical significance value was $p < 0.05$, and their correlation coefficient was >0.325. From the time domain features, the cross-correlation and autocorrelation features were shown to have statistical significance. Note that most of the cross-correlation and autocorrelation features exhibit multicollinearity indicating not all are required. Features obtained from peak velocities and peak-peak distance, specifically for the average of the two hands, were all shown to be informative features.

For the frequency domain features, ones extracted from the reshaped power spectrum showed the highest relevance. The average band power of the reshaped spectrum, average bandwidth in the reshaped spectrum, average peak height in reshaped spectrum, and the difference in band power of the reshaped spectrum were amongst the top features. This points to differences in power distribution of the frequencies in the “background of the motion” (i.e., other than the deterministic spectral lines) between individuals with (CON-S) and without a concussion (HC). Other frequency-domain features shown to be informative were the average value of F-test statistic for the first spectral line, the average band power, average power of first spectral line, difference in bandwidth and the difference in frequency value of the second spectral line. Note that the first spectral line is the 1 Hz frequency. As such, the F-test statistic for the 1 Hz frequency line and its power appear to be statistically significant. Also note that the second spectral line is any other deterministic line present in the signal (if any), and the difference between the right and left hand in the frequency value of this line is also shown to be significant.

Non-parametric univariate feature importance scoring was used to assess differences in the selected features given no prior assumptions about the data distribution. Very similar features were selected as the ones described above. Namely the features obtained from the reshaped spectrum were again selected here as the top features. In addition, features from the cross-correlation and the peak velocities were also selected, as well as the F-test statistic of the spectral lines. This method reinforces most of the features selected by the two previous univariate methods.

Finally, to account for the interactions among features and their overall influence on the performance of a predictive model for the response variable, a wrapper method was used. Since the genetic algorithm is stochastic in nature, it was executed three times using different random seeds thereby allowing the selection of different "optimal" feature combinations. Two types of fitness functions were used. One is to assess the ability to classify the samples into the two classes, i.e., hard classes. The second is a probabilistic predictions approach, wherein the probabilities are used rather than hard class assignments. Features were obtained from both methods and compared.

The features obtained from the reshaped spectrum were consistently selected as the best features, as well as the autocorrelation and/or cross-correlation features, which is again in agreement with the previous univariate methods. Main features from the PSD of the signal selected were the band power, median frequency, and the bandwidth. Coherence features were also occasionally selected but not consistently. Additionally, features that were also occasionally selected here were the slope and standard deviation of the instantaneous frequency. The features obtained from the average and difference between the right and left hands seem to be of equal importance. Note that the feature sets selected using the GA were often different, attributed to the random nature of the algorithm, however the different sets selected were generally of similar size and yielded similar performance scores. This is likely attributed to feature multicollinearity allowing highly correlated features to have very a similar impact on the response variable.

2.4 Binary Classification of Motor Impairment Following a Concussion

Using the features extracted in previous sections, supervised machine learning methods were implemented to explore the potential of classifying/identifying motor impairments indicative of a concussion. Since this is a proof-of-concept investigation, several machine learning algorithms were explored including linear and non-linear models. Linear models used were logistic regression, support vector machine with a linear kernel, and naïve bayes classifiers. Non-linear models used were random forest, Adaboost and XGboost classifiers, which are ensemble methods. Since the data has high dimensionality, dimensionality reduction methods were utilized. For the linear models, principal component analysis (PCA), recursive feature elimination (RFE), and L1 regularization (except for naïve bayes) were used as the dimensionality reduction/feature selection methods. As for the non-linear models, no dimensionality reduction techniques were implemented as tree-based algorithms have embedded feature selection procedures that reduce the dimensionality of the data as part of training. Nested cross-validation was used to select the optimal set of hyperparameters and test the selected model on unseen data. Two types of analyses were performed using the ML models: predicting hard classes and predicting probabilities. This section explores combinations of ML models and feature selection/reduction techniques that were utilized. Python (Scikit-Learn library for ML tasks) was used for all the work done in this section.

2.4.1 Supervised Learning Methods

The following section provides a synopsis of the models implemented for the classification task in this work. It is assumed that the reader has fundamental knowledge of the concepts discussed, for further information, refer to the book: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow [37].

Logistic Regression

Logistic regression (LR) is a linear model used for classification tasks. LR utilizes the logistic function with restricted maximum value, such that the model is bounded to the range [0,1]. Inherently, LR predicts probabilities, which are then converted to hard classes via a threshold. The threshold used in this work is 0.5, such that any sample resulting in an output probability >0.5 is classified as a concussion. LR does not require probability calibration as the model already outputs well calibrated probabilities. All the LR models in this work were regularized to reduce model complexity and the risk of overfitting. Models were regularized using the L2 penalty when PCA and RFE dimensionality reduction methods were used, and the L1 penalty when no other dimensionality reduction method was used. Both the L1 and L2 regularization parameters penalize coefficients/weights, however while L2 reduces the coefficients but keeps all the features, L1 shrinks coefficients of less important features to zero, thus acting as an embedded feature selection/reduction technique.

Support Vector Machine

Support vector classifiers (SVM) find the optimal hyperplane to separate the classes with the maximum margin. For this work, a linear kernel was used, and the SVM model constructed separates the classes using a linear boundary. Similar to LR, L1 and L2 regularization were implemented in the same manner. Furthermore, the same dimensionality reduction techniques were utilized as described above for LR. Contrary to LR, SVM outputs hard classes (i.e., 1 or 0), rather than probabilities. In order to obtain probability predictions, Platt scaling (sigmoid regressor) probability calibration is used. Platt scaling involves fitting a logistic model on the SVMs scores using an additional cross-validation procedure. While there are some theoretical issues with using Platt's scaling for obtaining probabilities from a SVM, results were still obtained in this work for comparison purposes.

Naïve Bayes

Gaussian naïve Bayes (NB) is a probabilistic classifier that is derived from Bayes theorem. It is naïve as it assumes that the features are independent, in that they do not influence each other, which is often violated. For this work a Gaussian likelihood function was used due to the continuous nature of the features. The raw predicted class probability estimates of the NB classifier are not reliable or reflective of the true class probabilities. To obtain more reliable probability estimates, the probabilities were calibrated using Platt's method. Note that Platt's method was used for all model calibration in this work because the alternative (isotonic) requires more data, and there is a limited number of samples available here.

Random Forest

A Decision Tree is a non-linear machine learning algorithm that splits the data continuously based on a criterion, until a condition is met, using the most informative features. Random Forest (RF) is an ensemble of decision trees that utilizes the idea of bagging or bootstrap aggregating to reduce variance. The RF model's prediction is the class with the greatest number of votes based on the decision trees outputs. RF offers improved predictions compared to a decision tree because it utilizes the idea of combining the output of various decision trees that have low correlation. Low correlation among the trees is maintained using two techniques. The first is bootstrap sampling of the training data, and the second is through random permutation of the features at every split resulting in potentially varying best splits. In this work, the minimum samples in a leaf, and minimum samples required to split an internal node were set to 2 and 6 respectively, to reduce the chance of overfitting. Tree-based models select the optimal set of features internally by selecting informative features to split the data at every node. This acts as embedded feature selection as the least informative features will not be selected (eliminated). Random forest models produce poorly calibrated probability estimates. As such, calibration using Platt's method was performed to obtain improved probability estimates. A simplified diagram describing the workflow of a random forest model is shown in figure 5 below [38].

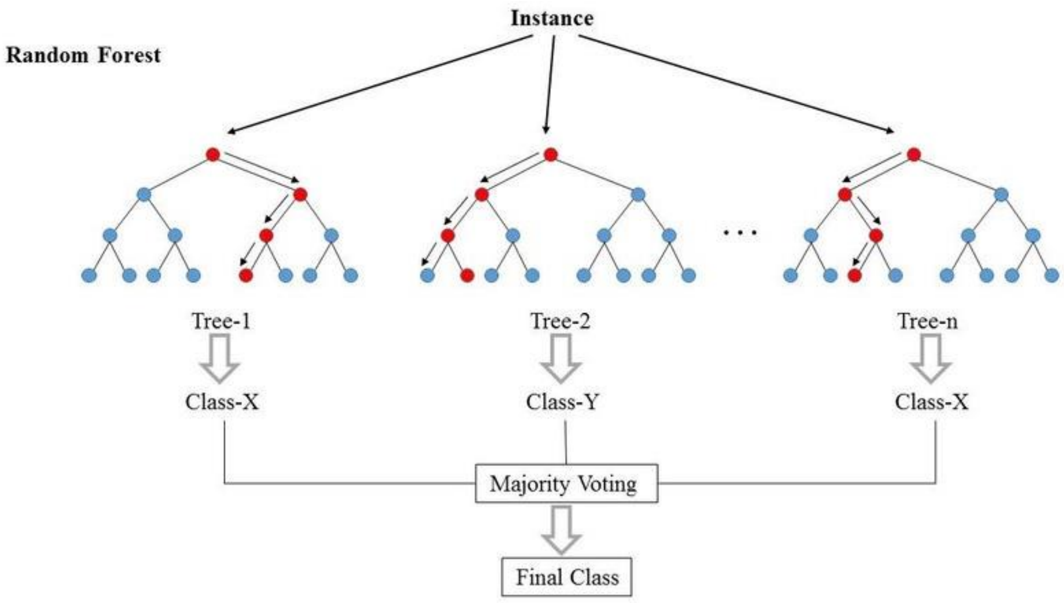


Figure 5: Simplified workflow of a random forest classifier [38]

Adaboost

Adaboost is boosting machine learning algorithm that combines an ensemble of weak learners to produce a strong classifier by reducing bias and variance. The weak learners used here are decision stumps, which are decision trees with a single split (depth=1). Contrary to bagging, boosting is sequential rather than parallel. In Adaboost, each weak learner is made to concentrate on data samples that were misclassified by the previous learner. This is done sequentially by altering the weights associated with the samples such that a higher weight is given to the misclassified cases in each iteration before training the next learner in the sequence. The prediction of the final classifier is then by weighted majority vote. For our dataset, Adaboost produced well calibrated probability estimates and therefore did not require calibration. Results are shown in the subsequent section. Figure 6 below shows a simplified illustrated of the process [39].

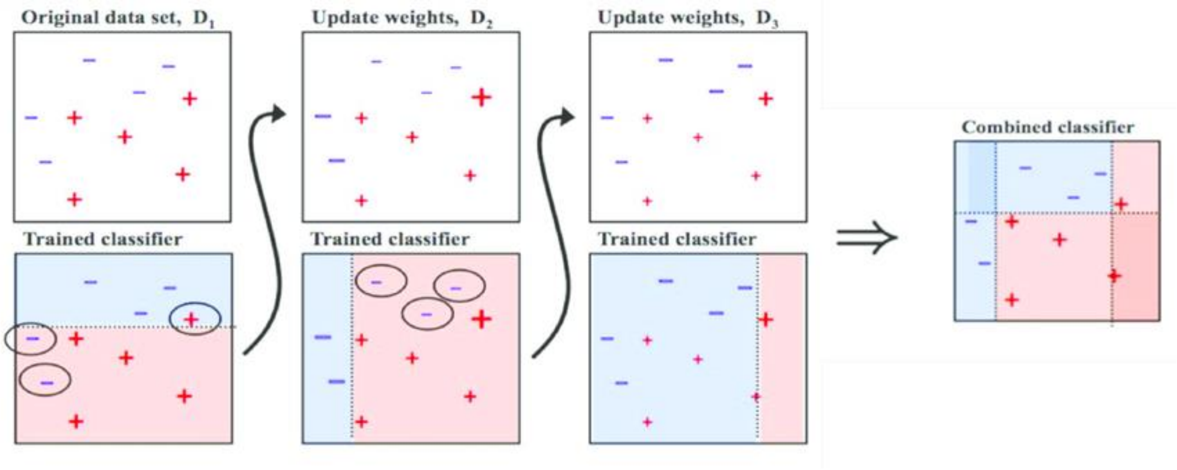


Figure 6: General depiction of the workflow of the Adaboost classifier [39]

XGboost

XGboost is another boosting technique that stands for extreme gradient boosting, aiming to reduce bias and variance. Gradient boosting works by sequentially training the subsequent weak learner with the residuals of the current learner, towards minimizing the loss function. XGboost is an optimized version of gradient boosting in terms of speed and efficiency. To reduce the chance of overfitting, XGboost can be regularized. For this work, L2 regularization was used. The probability estimates from XGboost were calibrated using Platt’s method. Figure 7 below summarizes the general workflow of XGboost [40].

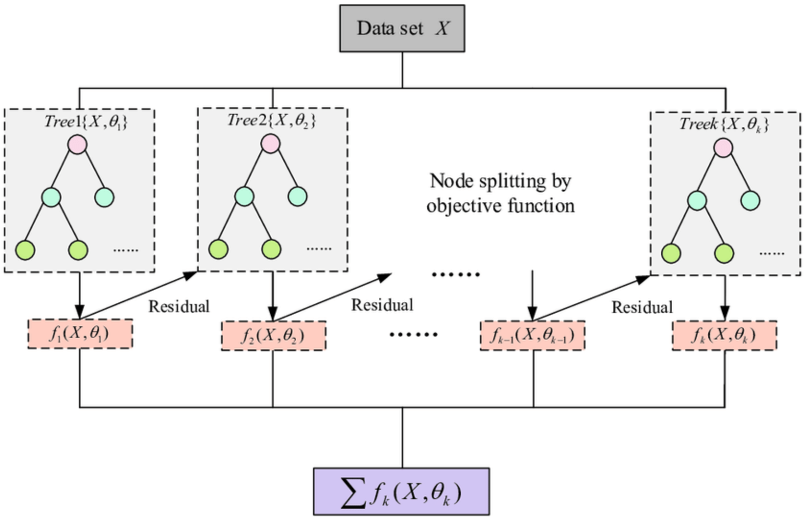


Figure 7: General depiction of the workflow of the XGboost classifier [40]

2.4.2 Dimensionality Reduction

Principal Component Analysis

PCA is an unsupervised learning method used for dimensionality reduction using variance as a measure of feature importance. PCA linearly transforms the feature set into a smaller set of orthogonal components/features that contains most of the variance (information) in the original features. The data was standardized prior to PCA. The number of principal components selected was chosen based on the amount of variance explained rather than the number of components directly (the percentage of variance in the data explained by the features).

Recursive Feature Elimination

Recursive feature elimination (RFE) is a wrapper feature selection method. RFE selects the optimal set of features by assessing the performance of an external model on the set of features. It recursively eliminates features deemed less important based on the feature weights computed by the external estimator, until the desired number of features is reached.

2.4.3 Performance Metrics

Accuracy

Accuracy is a measure of classification performance, that relies on hard class assignments rather than predicted probabilities, which is a drawback especially for evaluating probabilistic models. However, the accuracy metric is inappropriate for imbalanced datasets. The dataset used in this work for binary classification (HC vs CON-S) is balanced (19 and 18 samples per class), as such, class imbalance does not pose an issue on the metrics performance. For the multi-class classification case (HC vs CON-S vs CON-A), a balanced accuracy metric was computed, as the classes were no longer balanced (19, 18 and 11 samples per class). The balanced accuracy metric utilizes the true positive and true negative rates to compute the score. The equation below shows how the balanced accuracy was computed:

$$Accuracy = \frac{t_p + t_n}{t_p + f_n + t_n + f_p}$$

$$Balance\ Accuracy\ = \frac{1}{2}[\frac{t_p}{t_p + f_n} + \frac{t_n}{t_n + f_p}]$$

where $t_p, f_n, t_n\ and\ f_p$ stand for true positive, false negative, true negative and false positive respectively.

Brier Score

The Brier score (BS) is a proper scoring rule that assesses the probabilistic predictions of a model. BS computes the mean square error of the probability estimates with respect to the true values/classes. It can range in value from 0 to 1, wherein a smaller score indicates better model predictions. The following equation shows how the Brier score is computed:

$$Brier\ Score = \frac{1}{N_{samples}} \sum_{i=0}^{N_{samples}-1} (y_i - p_i)^2$$

where $y_i\ and\ p_i$ refer to the true value and the predicted probability respectively.

Mathews Correlation Coefficient

Mathew's correlation coefficient (MCC) measures the correlation between the predicted output and the true output. It uses the true and false positives and negatives to find the correlation coefficient in the range 0 to 1, and is not affected by class imbalances. A value closer to 1 indicates strong model predictive performance, a value near 0 indicates random model performance, while a negative MCC score closer to -1 implies inverse predictions. MCC provides a good summary of the confusion matrix using a single number [41]. The following equation shows how MCC is computed:

$$MCC = \frac{t_p \times t_n + f_p \times f_n}{\sqrt{(t_p + f_p)(t_p + f_n)(t_n + f_p)(t_n + f_n)}}$$

Area Under Receiver-operator Curve

The receiver-operator curve (ROC) is a plot of the true positive rate (sensitivity) over the false positive rate (1-specificity). The area under the ROC curve (AUROC) measures the ability of the classifier to distinguish or separate the classes. The higher the AUROC value, the better is the classifier's ability to separate the classes. An AUROC value of or near 0.5 indicates random classifier with poor classification ability while a value closer to 1 indicates stronger classification ability.

2.4.4 Method and Validation

In total, 6 classifiers and 3 dimensionality reduction techniques were implemented. The classifiers were validated using nested cross-validation (CV) to assess their performance on unseen data, and to assess appropriate machine learning models for this dataset that yield optimal performance. The purpose of the validation procedure implemented here is not to select the final model, but to compare performance, and assess the potential/feasibility of using the proposed kinematic data to build a useful model to perform the classification required for this task (i.e., identify concussion). As such, this study serves as an exploratory analysis.

In the nested cross-validation procedure, two CV loops were utilized. The inner CV loop's purpose is to optimize the hyperparameters and select the best features. This is performed using repeated stratified 17-fold CV with 10 repeats. The hyperparameter tuning process and feature selection is done using grid search whereby every possible combination of the chosen values for the selected hyperparameters are assessed using the CV procedure and given a performance score. The performance score is the average score for all folds and iterations of the CV. As such, one average score is given for each combination of parameters, and the model with the best score is selected. After all hyperparameters are assessed and the best performing model is selected, the optimal hyperparameters are used to fit a model on the training data of the outer CV loop.

The outer CV loop is a repeated stratified 18-fold CV with 30 repeats. The number of repetitions was selected to be 30 as larger number of repetitions was tested and did not yield significantly different results. Eighteen (18) folds were chosen for the outer CV loop because the dataset consists of 37 samples: 19 healthy and 18 with concussion. As such, 18 folds allows each fold to have one of each class, except for one fold which will have two from the healthy class and one from the concussion. The same reasoning applies for the inner CV loop. Each training fold of the outer CV loop contains ~35 samples (18 HC and 17 CON-S), which are then fed into the inner CV loop and further divided into 17 folds. Seventeen (17) folds were chosen so that most test folds will contain one sample from each class. The CV procedure was repeated in order to assess model stability, and to reduce the variance of the estimated predictive performance. In each repetition, the data was randomized before splitting the folds. Stability of the optimized hyperparameters was assessed visually by ensuring the hyperparameters selected for each CV loop did not vary drastically, and that the inner CV score was not drastically different from the test score on unseen data. Observed model instabilities are attributed to the small sample size and the potential presence of outlier(s) that is common with health/medical data.

In the outer CV loop, the training data was used to find features that had a correlation of 0.99 or more with other features, and they were eliminated. This was performed for every fold in the outer loop of the CV to ensure there was no information leakage if correlation was computed using the whole dataset. When PCA, LR, or SVM were used, the data was standardized by subtracting the mean and dividing by the standard deviation. After the performance metrics were computed using the results for each repetition/iteration, i.e.., for each full run through the dataset, they were appended and the results for each of the 30 repetitions were then used to compute the mean, and minimum & maximum values (range) for each metric. This was calculated to assess the overall performance of the models, and the variability in performance (for indirectly assessing stability). Figure 8 shows the general workflow implemented for training and performance assessment.

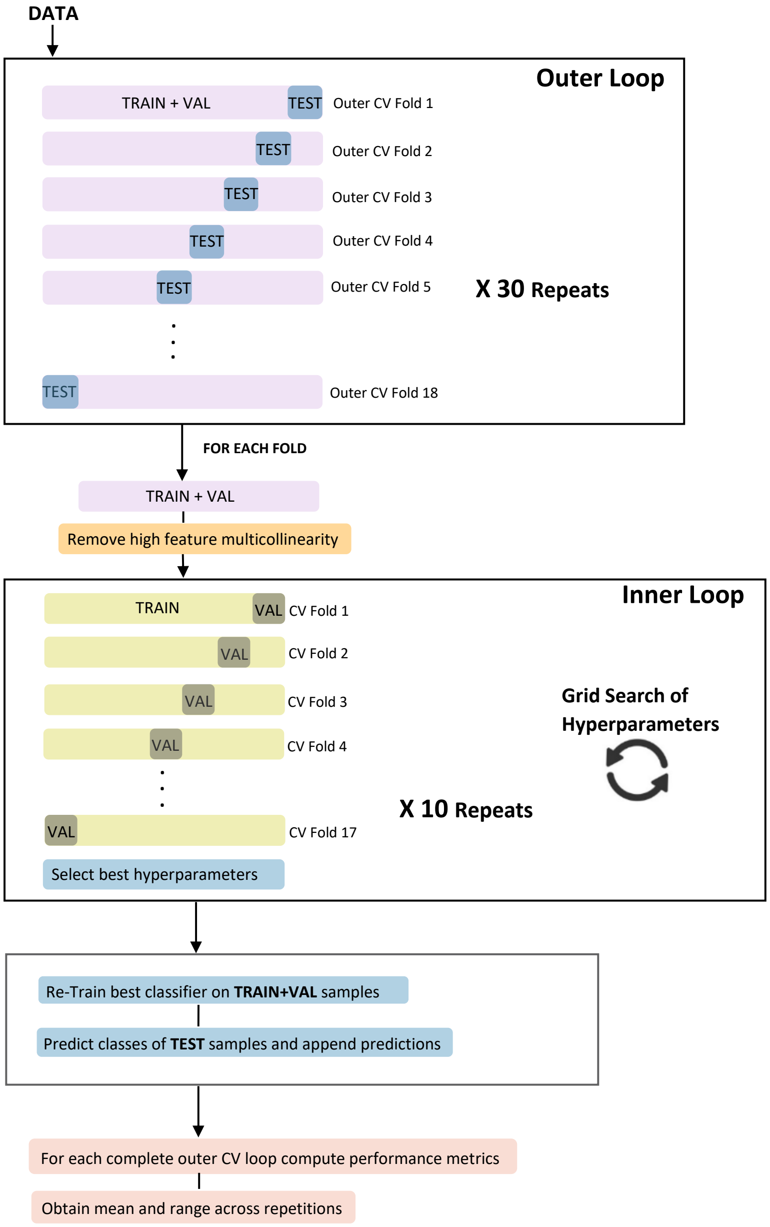

DATA
Outer Loop
TRAIN + VAL
TEST
Outer CV Fold 1
Outer CV Fold 2
Outer CV Fold 3
Outer CV Fold 4
Outer CV Fold 5
Outer CV Fold 18
X 30 Repeats
FOR EACH FOLD
TRAIN + VAL
Remove high feature multicollinearity
Inner Loop
TRAIN
VAL
CV Fold 1
CV Fold 2
CV Fold 3
CV Fold 4
CV Fold 17
Grid Search of Hyperparameters
X 10 Repeats
Select best hyperparameters
Re-Train best classifier on TRAIN+VAL samples
Predict classes of TEST samples and append predictions
For each complete outer CV loop compute performance metrics
Obtain mean and range across repetitions

Figure 8: Illustration of the cross-validation procedure for model selection and validation

2.4.5 Results

The results obtained from each of the discussed methods are listed below. Tables 8 and 9 show the results of the hard class predictions (0/1) performance evaluation. Tables 10 and 11 depict the performance scores of the probabilistic predictions. The mean accuracy, range of accuracy scores across repetitions, mean MCC score and the range of MCC scores across repetitions are shown to assess the hard classifications. As for the probabilistic predictions, the mean BS and range of BS across repetitions are tabulated. The mean of the AUROC and the range of the AUROC across repetitions are also included, and they assess the ability of the model to correctly rank the predictions (i.e., prior to thresholding) by using the model's probability estimates (or other score). Note AUROC does not assess the probability estimates produced by the model as it does not account for the predicted probability magnitudes.

Table 8 captures the results from the LR, SVM, and NB classifiers. Note that for NB, only PCA was implemented for dimensionality reduction, the second set of reported performance metrics are obtained from a NB classifier with no dimensionality reduction techniques. As a baseline, y-permutation (or scrambling) was implemented. In y-permutation, the true class labels of the training data samples are randomly permutated/resampled. Using logistic regression, the mean and range (min, max) for the accuracy of the trained baseline classifier was 40.3% (29.7 %, 48.6 %), and MCC score was -0.197 (-0.406, 0.033). Note that the negative in the MCC score implies inverse predictions. Using Adaboost baseline classifier, the average accuracy was 49.0 % (37.8 %, 64.9 %) , and the MCC score was -0.025 (-0.253, 0.296).

Table 9 shows the results obtained from tree-based methods, all of which were also ensemble methods (RF=bagging, Adaboost & XGboost=boosting). The best performing models are italicized and highlighted in the tables.

The performance metrics obtained from LR+PCA, SVM+PCA and NB were very similar. LR+RFE and SVM+RFE produced the highest scores obtained from the linear classifiers group. Overall, LR+RFE produced the highest average accuracy of 86.5% (MCC=0.75) followed by SVM+RFE with an average accuracy score of 82.1 % (MCC=0.666). In terms of the accuracy and MCC ranges produced over the repetitions, LR+RFE had the smallest range where every iteration had the same performance, followed by NB (10.8%, ΔMCC=0.22), and finally SVM+RFE (13.5%, ΔMCC=0.275). RFE selected very few features, mostly selecting just one feature for training the model, while PCA selected more features of up to 20.

Table 8: Results from linear classifiers (hard classes)

		Model		
FR	Metric	Logistic Regression	SVM	Naïve bayes
PCA	**Accuracy %**	75.0 (67.6, 83.8)	74.6 (62.2, 83.8)	65.8 (56.8, 78.7)
	MCC	0.503 (0.351, 0.678)	0.495 (0.242, 0.678)	0.318 (0.138, 0.518)
RFE *None* *(NB)*	**Accuracy %**	***86.5 (86.5, 86.5)***	82.1 (73.0, 86.5)	76.7 (70.3, 81.1)
	MCC	***0.756 (0.756, 0.756)***	0.666 (0.481, 0.756)	0.543 (0.408, 0.628)
L1	**Accuracy %**	63.8 (58.3, 69.4)	71.5 (58.3, 77.8)	-
	MCC	0.286 (0.169, 0.422)	0.433 (0.167, 0.570)	-

For the ensemble tree-based algorithms, boosting methods outperformed bagging. All the ensemble tree-based methods performed better without any external dimensionality reduction techniques. The highest performance metrics obtained here were for the Adaboost classifier with an average classification accuracy of 88.9% (MCC=0.795), followed by XGboost with an average accuracy of 88.8% (MCC=0.793). In terms of the accuracy and MCC ranges produced over the repetitions, Adaboost had the smallest range (2.7%, ΔMCC=0.064), followed by XGboost (5.4%, ΔMCC=0.124), and finally by RF (17.3%, ΔMCC=0.124).

Table 9: Results from ensemble tree-based classifiers (hard classes)

	Model		
Metric	Random forest	Adaboost	XGboost
Accuracy %	85.9 (83.8, 89.2)	***88.9 (86.5, 89.2)***	88.8 (83.8, 89.2)
MCC	0.725 (0.678, 0.802)	***0.795 (0.738, 0.802)***	0.793 (0.678, 0.802)

As for the probabilistic predictions, the baseline y-permutated logistic regression model yielded a BS of 0.265 (0.242, 0.292) and an AUROC value of 0.405 (0.327, 0.478). The Adaboost baseline model produced a BS of 0.294 (0.248, 0.330) and AUROC of 0.496 (0.363, 0.636). Naïve Bayes resulted in poorly calibrated probabilities, despite calibration. As such, its results were not included as it was deemed a poor classifier for this dataset. From the linear models, RFE+LR yielded the best performance with a BS of 0.117 (AUROC=0.879), and a range across repetitions of 0.047 for the BS (0.091 for the AUROC). The reliability curves (calibration curves) for the LR and the SVM models are shown in Figures 9 and 10 below. A calibration curve was produced and inspected for each repetition. However, all the outputs were pooled and used to produce one overall calibration curve for the purpose of illustration here. The average calibration curves did not vary significantly from individual calibration curves for any of the classifiers tested.

Table 10: Results from linear classifiers (probabilities)

		Model	
FR	Metric	Logistic Regression	SVM
PCA	**BS**	0.191 (0.141, 0.226)	0.196 (0.169, 0.230)
	AUROC	0.791 (0.734, 0.857)	0.784 (0.675, 0.865)
RFE	**BS**	***0.117 (0.113, 0.118)***	0.14 (0.118, 0.170)
	AUROC	***0.879 (0.874, 0.892)***	0.856 (0.788, 0.889)

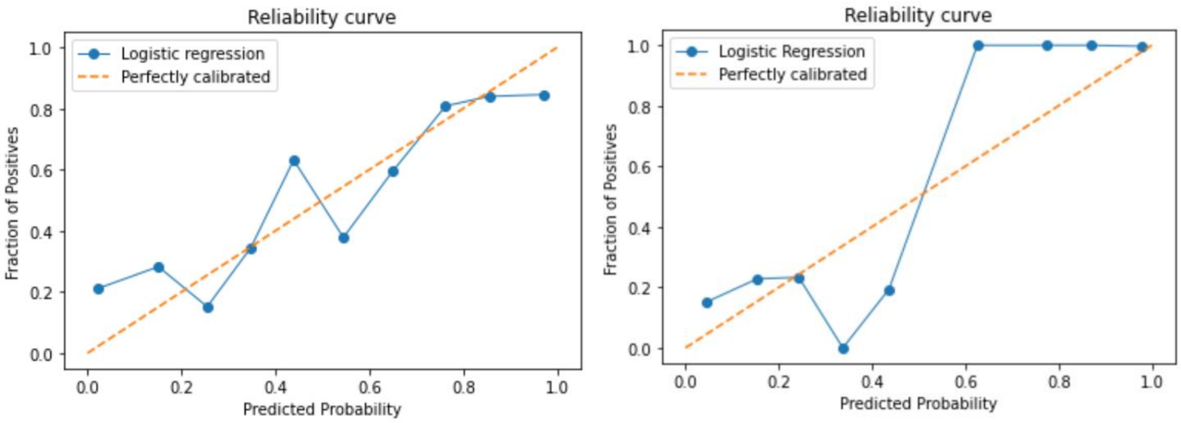


Figure 9: Reliability curve for the logistic regression model showing the true frequency of the positive classes against the predicted probabilities using PCA (left), and RFE (right)

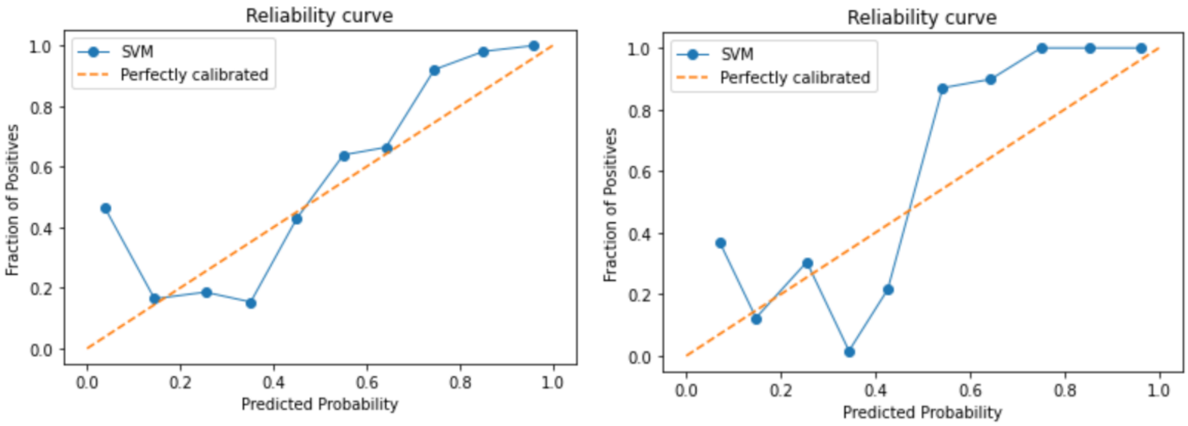
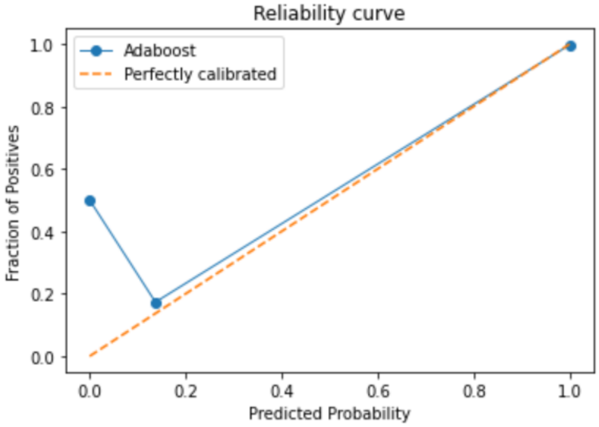


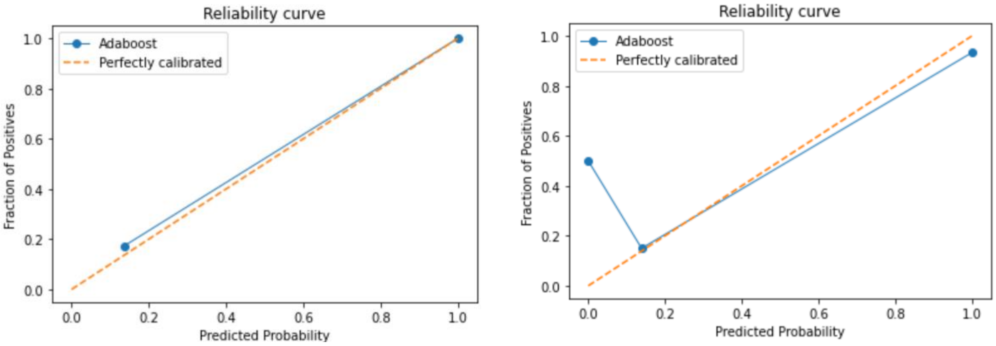
Figure 10: Reliability curve for the support vector machine classifier showing the true frequency of the positive classes against the predicted probabilities using PCA (left) and RFE (right)

Table 11 below depicts the probabilistic performance of the ensemble tree-based classifiers. Adaboost generally outperformed other classifiers with a BS of 0.096 (AUROC=0.821), and range across repetitions of 0.032 (AUROC=0.006). In addition, the model already produced very well calibrated predictions and did not require calibration. Figure 11 shows the reliability curve for the Adaboost classifier along with the number of samples in each bin. Only 1 of the 1110 predictions produced from one of the 30 models caused the first bin to diverge from the perfectly calibrated line. This was due to one of the two predicted samples (in one bin) being misclassified with high confidence, resulting in the fraction of positives being 50% when the sample was confidently predicted as negative by the model. Figure 12 depicts the calibration curve for all the other repetitions (29 total) produced. Adaboost was confident in its predictions such that it produced probabilities that were closer to 0 and 1, rather than midrange.



```
array([  2, 687,    0,    0,    0,    0,    0,    0,    0, 421,    0], dtype=int64)
```

Figure 11: Reliability curve for the Adaboost classifier showing the true frequency of the positive classes against the predicted probabilities, along with the number of samples in each bin



```
array([ 2, 20,  0,  0,  0,  0,  0,  0,  0, 15,  0], dtype=int64)
```

Figure 12: Right: Reliability curve for the one model containing the confidently misclassified sample. Left: Reliability curve from one of the other models representing the other 29 models.

RF and XGboost yielded poorly calibrated outputs, as can be seen from their reliability curves in Figure 13, despite classifier calibration using Platt’s method. A larger sample size is required for potentially improved calibration using isotonic regression.

Table 11: Results from the ensemble tree-based classifiers (probabilities)

	Model		
Metric	Random forest	Adaboost	XGboost
BS	0.132 (0.126, 0.141)	***0.096 (0.094, 0.126)***	0.144 (0.130, 0.159)
AUROC	0.787 (0.778, 0.798)	***0.821 (0.820, 0.826)***	0.786 (0.763, 0.798)

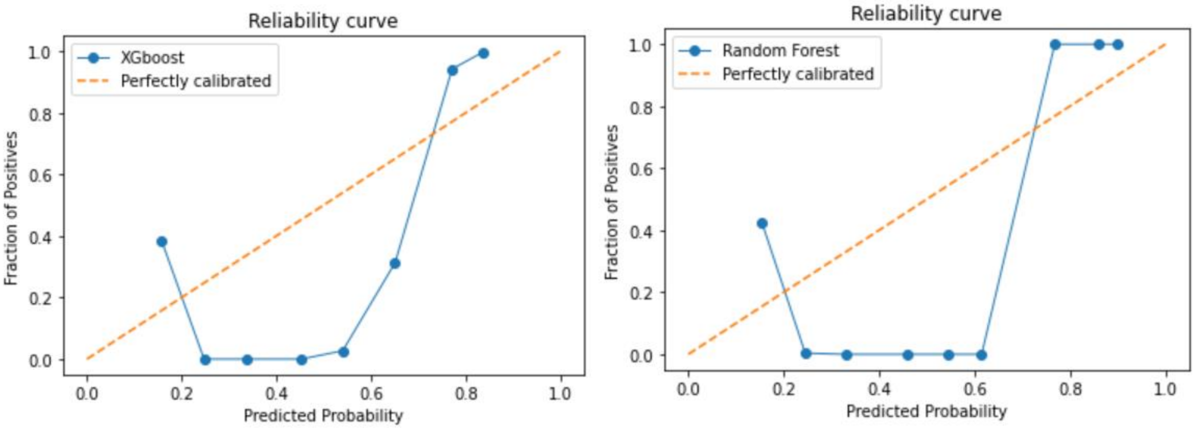


Figure 13: Reliability curves for the XGboost (left) and the random forest (right) classifiers

2.5 Multiclass Classification of Motor Impairment Following a Concussion

For the multiclass classification case (HC vs CON-S vs CON-A), the asymptomatic group (CON-A) was included in the analysis to determine if the asymptomatic group can be classified separately or if they resemble one of the other two groups. Which group they resemble the most is also important to determine if this model is able to detect a concussion even when the injury was not recent and/or the individual is asymptomatic, or if the model is only able to detect individuals with a recent injury and ongoing symptoms. Note that this analysis is not conclusive due to the small sample size available for the asymptomatic group (n=11), however it serves as a preliminary analysis.

Two models were used for the multiclass classification task: Random Forest, and XGboost classifiers. The RF and XGboost classifiers are both inherently multiclass. Only hard class predictions were performed with the multiclass models. The metrics used are the same as those described in the previous section. No hyperparameter tuning was performed, default parameter settings were kept. However, for both models, different weights were given to each class as per the class frequencies. Note that for the AUC-ROC measure obtained here, the one vs. one classification method was used.

2.5.1 Method and Validation

For the multiclass classification, no hyperparameter optimization was performed and leave-one-out cross-validation (LOOCV) was employed as the cross-validation scheme for assessing performance on unseen data. The general workflow of a LOOCV is shown in Figure 14 below wherein the shaded region is the test fold, and the white region is the training fold [42]. These changes were made because the asymptomatic class has a smaller dataset of n=11 samples. In addition, the dataset is not balanced, so the models and the assessment metrics were adjusted accordingly by accounting for the class frequencies.

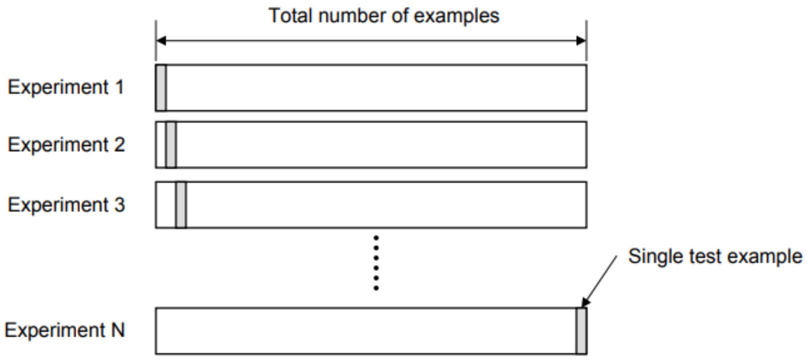


Figure 14: LOOCV where one sample is tested in each fold (experiment), and N-1 samples are used for training. There are as many folds as there are samples [38].

2.5.2 Results

The results obtained for the multiclass classification task are shown in Table 12 below. XGboost resulted in an accuracy of 70.3% , MCC score of 0.582 and an AUROC of 0.809. Tables 13 and 14 show the classification report for the RF and XGboost classifiers respectively. Note that class 0 represents the healthy class while classes 1 and 2 are the symptomatic and asymptomatic concussion classes respectively.

Table 12: Results from the multiclass classification task

	Method/model	
Metric	XGboost	Random forest
Accuracy %	70.3	62.6
MCC	0.582	0.481
AUROC	0.809	0.765

Table 13: Classification report for the random forest multiclass classifier

	precision	**recall**	**f1-score**
class 0	63.6	73.7	68.3
class 1	73.7	77.8	75.7
class 2	57.1	36.4	44.4
macro avg	64.8	62.6	62.8
weighted avg	65.9	66.7	65.6

Table 14: Classification report for the XGboost multiclass classifier

	precision	**recall**	**f1-score**
class 0	69.6	84.2	76.2
class 1	76.5	72.2	74.3
class 2	75.0	54.5	63.2
macro avg	73.7	70.3	71.2
weighted avg	73.4	72.9	72.5

As can be seen from the classification reports for both the classifiers, poorest performance was generally obtained for class 2 (asymptomatic, CON-A). The XGboost outperformed RF with a weighted average for precision, recall and f1-score of 73.4 %, 72.9 % and 72.5 % respectively. Note that the precision is the ratio of true positives to all positives, recall is the true positive rate (or sensitivity) discussed previously and the f1 score is the harmonic mean of precision and recall.

When comparing the asymptomatic group individually with each of the healthy and the concussion groups, it showed higher resemblance to the healthy group and more separability from the concussion group. Using a RF model, the scores for accuracy, AUC-ROC and MCC for the asymptomatic-healthy classifier were 53.8%, 0.675 and 0.109 respectively. On the other hand, also using a RF model but for the asymptomatic-concussion group, the scores were 90.9%, 0.838 and 0.858 for the accuracy, AUC-ROC and MCC metrics respectively.

2.6 Binary and Multiclass Classification Discussion

Overall, the findings from this study demonstrate a moderate-to-strong ability to detect individuals with a concussion using kinematic metrics of bimanual coordination. Results from this study indicate the potential to detect concussion with up to 88% accuracy (AUROC= 0.82) using an Adaboost classifier, and 86% accuracy (AUROC=0.87) using a logistic regression model, with the proposed collection paradigm. Results also suggest that asymptomatic individuals (with a history of concussion) did not exhibit the same motor deficits as symptomatic individuals (with a recent diagnosis). However, it is recognized that these results are preliminary and obtaining a larger sample size is necessary for more definite conclusions. The reason for testing on different algorithms was to assess the separability of the classes given various criteria and models.

While the results agree with the first hypothesis posed in this study, they disprove the second. Very few (sometimes only 1) feature(s) were selected by most classifiers, which were shown to have strongest relationships with the outcome. This is contrary to the hypothesized outcome wherein it was predicted that there would be small contributions from several features. In particular, power-related features were shown to be the strongest predictors. The band power, power of the 1 Hz line, and the background spectrum total power were all higher for the concussion group (CON-S), compared to the healthy group (HC). Since a single feature was selected by some classifiers, this points to most of the features discovering the same pattern(s), such that when each feature is used separately, they yield better results than when combined (leading to overfitting). This result was surprising but points to the potential usefulness of simplistic models for concussion identification using this task.

The best performing classifiers for this dataset were logistic regression and Adaboost. In terms of the best calibrated probabilities, Adaboost outperformed logistic regression. For logistic regression, using PCA produced slightly better calibrated probabilities than RFE, although the predictive performance using RFE outperformed PCA significantly. Note that logistic regression with RFE and Adaboost both chose minimal number of features (mostly selecting just 1 feature), while PCA feature reduction utilized more features (~20 features), as well as random forest (~25 features).

Considering not all concussions affect bimanual coordination, depending on the region of brain damage, not all participants in the concussion group in this study were expected to show significant motor impairments. As such, accuracy results obtained by the proposed classifiers appear reasonable and appropriate evidence supporting a bimanual coordination screening paradigm. With the potential to provide more sensitive, objective sideline concussion screening, this paradigm could reduce missed diagnoses and inappropriate return to play decisions.

3 Low-Cost Portable Kinematic Measurement System for Bimanual Coordination Assessment

3.1 Background

Despite the existence of standardized tests for assessing several motor tasks (e.g., mBESS), there lacks a standardized procedure for assessing bimanual motor coordination. In the previous chapter, a novel motor coordination assessment paradigm was proposed and shown to be effective in identifying coordination deficits associated with concussion among athletes. However, such a testing procedure requires motion capture cameras for data collection. This diminishes its feasibility for use on the sidelines of a sporting event. While motion capture is the gold standard for motor assessment, it requires expensive equipment and must be performed in a laboratory setting. This chapter proposed an alternative method for collecting the data discussed in Chapter 2 that is portable and feasible for use as a quick screening tool.

Common portable devices for motor assessment are wearable sensor technologies, such as inertial measurement units (IMUs). Wearable sensors have been used for assessment of several motor tasks for concussion screening such as gait, and balance [43,44]. Another quantitative method that has been used for concussion screening is applications that employ smartphone sensors for motor assessment, specifically of balance [45]. However, there has not been any reported work on objective tools for assessing bimanual motor coordination that could effectively and feasibly be used for concussion screening on the sidelines.

Currently implemented methods for bimanual motor assessment on the sidelines for concussion screening rely on subjective methods that are based on human judgment. For the proposed experimental paradigm in this work, position data is collected to provide data-driven decisions regarding return-to-play. Data collection systems that can be used as alternatives to motion capture, and are portable and feasible, include ultrasonic sensors, infrared sensors, and RGB cameras. While the ultrasonic sensor would permit portability and low equipment cost, it still requires the use of specialized hardware (although simple). Furthermore, issues persist with regards to inconsistencies in sampling rate, and inconsistencies in performance/accuracy. A study conduced by our research group assessing the performance of ultrasonic sensors for the proposed paradigm showed that the results obtained were overall inconsistent, and not accurate enough for the system requirements. Moreover, the evaluation demonstrated that coloured marker tracking (using video) outperformed the ultrasonic sensor data [46]. While infrared sensors are also cost effective and portable, they are not always reliable and are influenced by environmental conditions, especially in outdoors settings attributable to interference from sunlight. In addition, they also require specialized hardware. This makes them less feasible for rapid use on the sidelines. A more robust and readily available sensor is a camera (RGB light sensor). A camera is readily embedded in all modern smartphones, eliminating the need for additional hardware.

The method selected for motion tracking is video processing/computer vision. Video processing methods, namely coloured marker tracking, will be used as baseline for the markerless computer vision system due to several undesirable attributes of object tracking which significantly reduce reliability, ease of use, and feasibility. Furthermore, one of the requirements set forth for this design was to minimize the number of components required for data collection/use. As such, the focus of this chapter will be on computer vision methods for markerless hand tracking, to collect the required position data. There have been significant recent advances in computer vision systems, and an increase in their use for several applications. While there have been no reports of such tools being used for bimanual coordination assessment, there have been reports of its use for other biomechanical assessments such as gait and balance [47,48,49]. For gait assessment using computer vision, one work reported the use of five video cameras sampling at 30 fps (4K resolution) and 120 fps (1K resolution) to obtain an accuracy of ~30 mm using OpenPose, for 3D pose tracking [49]. In another work, two HD webcams sampling at 30 fps (1K resolution) were used to obtain an accuracy of 20 mm for 3D pose tracking using OpenPose, also for gait analysis [48]. Such tools allow for motor analysis outside of a lab setting for several applications including concussion screening. However, their use for assessing motor skills other than balance and gait has not been explored.

This chapter will focus on the design of a data collection tool for assessment of bimanual motor coordination, specifically of the arms. The objective is to develop a low-cost system for rapid data collection on the sidelines of sporting events.

The research questions posed in this chapter are: Can a computer vision system be used to obtain reliable metrics for quantifying performance on upper body bimanual coordination tasks? Are the metrics obtained through the computer vision system of comparable accuracy to the motion capture gold standard?

3.2 Design Requirements

Design requirements based on a sideline screening application must be met for the system to be deemed feasible for the required use. Some of the requirements are technical, others are usability conditions. The targeted users are sports clinicians and coaches. Table 15 outlines the constraints of the design.

Table 15: Design constraints for the data collection system

Constraint	**Requirement**
Rapid	Takes no longer than 30 minutes to provide results
Low-cost	Readily available components, no external devices required for purchase
Mobile	Can be used outside of a lab setting, specifically on the sidelines of sporting events
Accurate	Maximum pk-pk error of 37 mm compared to research grade motion capture, and maximum peak velocity error of 131 mm/s
Robust	Functionality not heavily influenced by environmental/external factors

Since the system is to be used on the sidelines of sporting events, it must be mobile so that a clinician/coach can readily use it in the event of a suspected concussion. It must provide feedback in a short enough time frame such that the result can inform return to play decisions. A maximum time frame of 30 minute was chosen as an initial specification. For the system to be implemented and used by the targeted users, it must be low cost and require no heavy/large components or devices be carried. In terms of technical constraints, the system must be accurate enough to produce useful results. The maximum allowable peak to peak (pk-pk) error is 37 mm, and the maximum allowable peak velocity error is 131 mm/s, based on motion capture data. The allowable error values were selected by performing statistical tests, which will be discussed in following sections. While lower error values are required in practice (<10 mm), these values provide a target specification for assessing general performance. Note that good pk-pk accuracy is of higher importance than the maximum peak velocity. While the pk-pk error illustrates the accuracy of position features, from which other features are also derived, the peak velocity is a selected feature that can be replaced by another potentially more or equally informative feature. Finally, the system must be robust in the face of external environmental conditions such that minor changes in lighting, weather, background, skin colour, and location do not disrupt its usage. The criteria would be to optimize the listed constraints so as to reduce cost, increase accuracy and speed. An additional criterion is ease of use with minimal required effort for data collection. A complicated system or one with several procedural steps will likely be misused or unused.

Given the discussed requirements, it was decided that a smartphone will be used as the data collection device as it is readily available to everyone, and easy to use without specific training. The phone's camera was selected as the sensor to be used for data collection, and computer vision will be utilized to obtain the data of interest from the collected video.

3.3 Methods

Given the specific requirements for the data collection apparatus, all design, testing and selection criteria will be based on settings outlined in Chapter 2,. Three different methods were prototyped, tested, and analyzed: 1) coloured marker tracking, 2) hand bounding box, and 3) hand landmarks tracking. The data used for decision making is based on finger tracking (outlined in Chapter 2). However, given the 2D nature of an image, accurate finger tracking is a complex task. As such, initial methods were used to track a hand fist instead, which was deemed an acceptable alternative, as it was assumed to produce the same patterns in the data as finger tracking. Colour marker tracking and bounding box were therefore used for hand fist tracking. As for the hand landmarks tracking, it was further divided into two parts: a) hand fist tracking and b) finger tracking at an angle.

Coloured Marker Tracking

Colour tracking of a yellow marker was implemented as a marker-based video processing method for hand tracking. Colour tracking was implemented on MATLAB using Algorithm 1 described below. Due to the lack of a standardized distance, surrounding and lighting conditions, this method required some manual parameter adjustment for some trials due to sensitivity to environmental conditions. Note that this method could potentially be further optimized to improve speed and detection (utilizing more advanced blob detection algorithms). Improvements were not implemented in this work as this method was not desirable for the given application, and markerless methods were preferred.

Algorithm 1 Colour Marker Tracking

```
Input: Trial video
Output: Pixel position data
While video_frames_available == True:
    ImageY←create yellow mask
    ImageF←Filter ImageY based on object radius, area, and pixels in its neighbourhood
    Centroid←Find the centroid of each detected object
    Left and right hand positions ←Determine handedness based on location of the centroid
```

Tracking Hand Bounding Box- HandTrack.JS

The first markerless alternative that was implemented was hand fist tracking using a bounding box. This was achieved using the Handtrack.JS JavaScript library, which is intended for real-time hand detection, in a web browser [50]. To use an existing video as input to the model, an additional script was written to feed in individual video frames and append the output. This library provides a trained convolutional neural network using frontal view hand images, which is the same perspective as the desired fist tracking task.

Tracking Hand Landmarks- MediaPipe Hands

MediaPipe Hands is a graph-based framework which consists of a machine learning pipeline wherein multiple models are used together to detect and track hand landmarks [51]. The method first implements a palm detection model then utilizes the output to localize a region for the hand landmarks model, yielding better performance. In addition, it is able to detect and track more than one hand in an image frame. Mediapipe was implemented in C++ for this work; however currently there are Mediapipe APIs in Python, JavaScript, Android and iOS.

Mediapipe detects 21 hand landmarks, which are located on the knuckles of the hand. Figure 15 below shows the location of the landmarks.

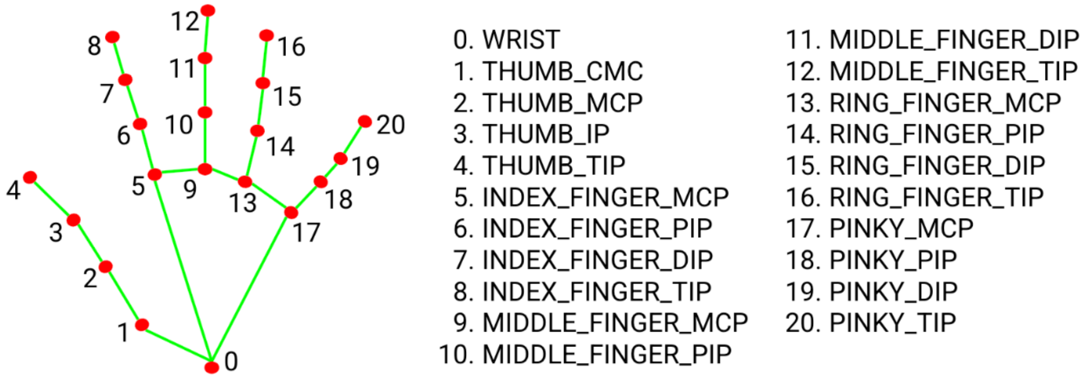


Figure 15: Location of the hand landmarks detected and tracked by MediaPipe

Hand Fist: For hand fist tracking, landmark 5 was used, which represents the index finger knuckle.

Finger: For finger tracking, landmark 8 was used, which represents the tip of the index finger.

Calibration

Proper calibration is essential to obtain accurate measurements. The calibration procedure needs to be easy and rapid to satisfy the design constraints. Additionally, calibration must be automated and require no manual work or user input.

To calibrate the camera, such that pixel values are converted to real world distances, an initial short calibration video was taken prior to the data collection trial. The calibration video must be captured using the same camera, position, framerate, and resolution settings as the test. Calibration is performed using a small checkerboard pattern; however, unlike camera calibration, the user will not be required to take multiple pictures or move the checkerboard around the view. The checkerboard pattern is instead used as a reference distance. This pattern was used for calibration as finding corners of the squares can be done automatically to obtain pixel distances, and the real square sizes are known a priori. Two checkerboard patterns were used, one for each hand, each of which is made of 5x4 squares, as shown in Figure 16 below. The pattern is attached to a cardboard to ensure it is rigid then a finger glove is glued to the back to allow the user to wear it during calibration.

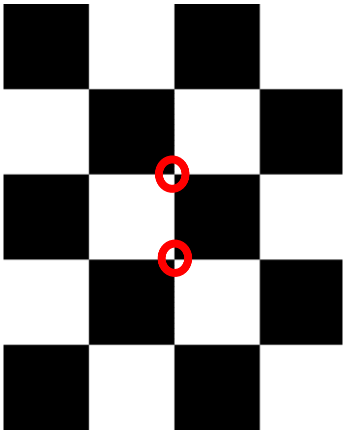


Figure 16: Checkerboard pattern used for calibration, with the corners used for determining the pixel length circled in red, which corresponds to a real vertical distance of 20.4 mm

To calibrate for the hand fist trials, the participants were asked to place the glove between their fingers and move their hands vertically, similar to the actual trial, for about 5 seconds such that the same space is covered as during the collection trial. For finger tracking, the participants were asked to wear the finger glove on their index finger and move their arms in the same manner.

The steps for obtaining the calibration ratio from the calibration video are described in Algorithm 2 below. Steps 1-4 were also used for obtaining the calibration curve which will be discussed in a subsequent section. Each of the left and right hands must be calibrated, and a ratio obtained for each hand. From Figure 16 above, the circled corners were tracked and used for calibration. Several assumptions were made for calibration. First, this procedure assumes that the pixel length of the square is constant regardless of vertical position (neglecting changes due to camera orientation/perspective). Second, since the shoulder has a spheroid joint, the vertical movements of the arm will not produce a pure vertical motion, as such, at the top and bottom of the motion, the real pure vertical distance of the squares will be slightly different than the constant value used. This difference is assumed to be negligible. In addition, the algorithm for detecting the checkerboard pattern is more sensitive to background noise than the hand tracking. As such it is recommended to keep the background less noisy when possible, by reducing objects in the background, specifically objects with many corners.

Algorithm 2 Calibration Constant Estimation

```
    Input: Calibration video
    Output: Calibration ratio
1:  While video_frames_available == True:
    1:  Crop image frame to obtain right or left hand calibration and get rid of unneeded regions
    2:  Pixel distances ← Detect checkerboard points
    3:  If detected corners<12:
        1:  Delete distance vector for this frame
    4:  Concatenate pixel distance vector vertically
2:  For i=5:12:length(pixel distance vector)
    1:  Pixel square length= pixel distance vector (i) - pixel distance vector (i+3)
    2:  Concatenate Pixel square lengths
3:  Filter outliers + median filter
4:  Find Peaks and Valleys in the filtered pixel square lengths vector
5:  Mean pixels = mean of the peaks and valleys
6:  Calibration ratio=Real square length/ Mean pixels
```

3.4 Analysis of Different Methods

The 4 studied methods vary in their advantages and disadvantages. The methods were selected on a basis of increasing complexity. Initially the simplest method, colour tracking, was implemented and tested. Following that, markerless tracking was examined with the least complex bounding box tracking method. Finally, the most complex method, landmark tracking, was implemented and tested. Table 16 below summarizes key advantages and drawbacks of the examined methods. Execution times listed in the table below are the average time to run on an Intel Core i7 with 2.90 GHz clock speed and 64-bit windows operating system.

Table 16: Comparison of the different tracking methods considered

Method	**Analysis**
Coloured Marker Tracking	***Advantages:*** *- High accuracy* ***Disadvantages:*** *- Highly influenced by background and surrounding colour and lighting conditions* *- Higher processing time (~30 minutes for 20 seconds 60 fps video)* *- Often require parameter adjustment as per trial conditions therefore potentially requiring manual work* *- Markers required, as such, more components needed for data collection*
HandTrack.JS	***Advantages:*** *- Easy to use, no markers required* *- Faster processing time (~15 minutes for a 60 fps 20 seconds video)* *- More robust, results are not influenced by background and surrounding conditions, or participant's clothing* *- No manual adjustments required, automatic results* ***Disadvantages:*** *- Low accuracy and precision* *- Several misclassifications*
Mediapipe - Fist	***Advantages:*** *- Good accuracy* *- Easy to use, no markers required* *- Faster processing time (~15 minutes for a 60 fps 20 seconds video)* *- More robust, results are not influenced by background and surrounding conditions, or participant's clothing* *- No manual adjustments required, automatic results* ***Disadvantages:*** *- Accuracy still not as good as motion capture or colour tracking*
Mediapipe - Finger	***Advantages:*** *- High accuracy* *- More closely depicts the experimental procedure for the dataset of interest* *- Easy to use, no markers required* *- Faster processing time (~15 minutes for a 60 fps 20 seconds video, ~20 minutes for a 120 fps 20 seconds video, and ~30 minutes for a 240 fps 20 seconds video)* *- More robust, results are not influenced by background and surrounding conditions, or participant's clothing* *- No manual adjustments required, automatic results* ***Disadvantages:*** *- Accuracy negatively affected by motion blur at low framerates (60 fps or less)*

To test and compare their performance, prototype algorithms were generated and tested against Vicon motion capture (Vicon Vantage V5 cameras) data as reference. An iPhone XR was used for all video data collection, with a video framerate of 60 fps and 4K resolution.

3.4.1 Testing Procedure

Data from two participants was used to test the performance of the four methods. The apparatus for data collection is shown in Figure 17 below. A physical guide with protruding markers (180 mm apart vertically) was placed on the right-hand side of the participants to guide their motion. They were instructed to remain within these makers as boundaries. The participants were asked to sit on a chair with a table in front of them, on which the physical guide was placed. For the auditory guide, a 1Hz monotone beep was used during the trials and the participants were asked to synchronize their motion such that they produced 1Hz vertical arm movements. For three of the methods, colour tracking, bounding box, and fist landmarks, participants were instructed to maintain their hands in a fist position with one retroreflective marker placed on the side of each fist, and one colour marker placed on a finger in each hand. The smartphone was placed on a tripod in front of the participant at a sufficient distance to capture the entire movement. Slight angles or changes in perspective were unintentionally introduced between trials and were not strictly measured as these changes would be expected in real life usage. For calibration, a 5x4 checkerboard pattern was used, as that is the minimum size required for automatic detection and tracking. The two checkerboard patterns were then placed between the fingers to hold it in place and the participant was asked to perform the vertical arm motion with the checkerboard pattern.

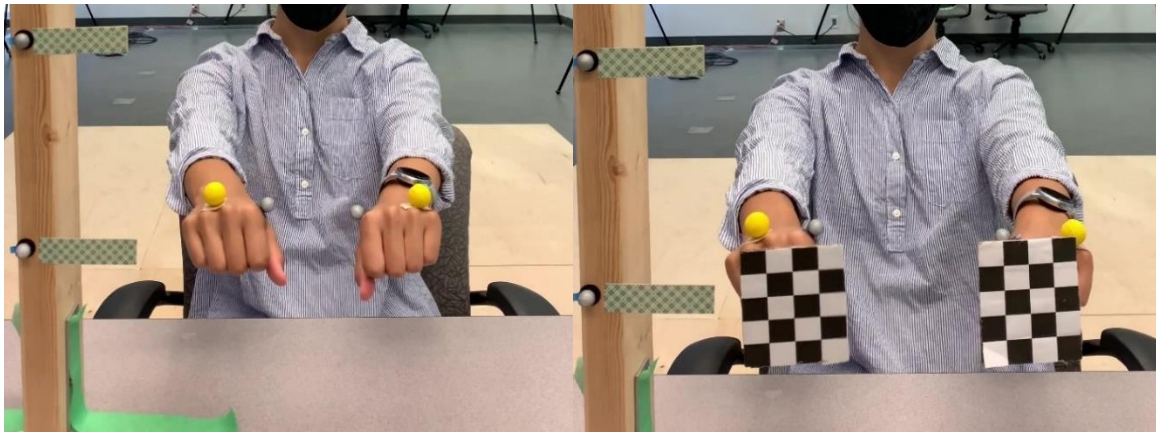


Figure 17: Left: Hand fist tracking experimental setup. Right: Snapshot of calibration video

For the finger landmark tracking, the participant was required to extend their index finger and a retroreflective marker was placed on the side tip of the index finger of each hand. The camera placement in this procedure was different than the hand fist method, such that the camera was placed at an angle to the participant's left side. Figure 18 shows the camera's view for this procedure. Similar to the previous methods, the exact angle and orientation were not measured. As long as the two fingers were fully observable and did not overlap each other, the tracking was overall successful. The guide was to place the camera at about a 45 degrees angle relative to the sagittal plane. This angle was selected based on experimenting with different angles/settings to find the optimal arrangement. In addition, the camera was slightly tilted downwards for all trials. Calibration was performed in the same manner as the fist trials; however, the checkerboard pattern was worn on the index finger using the finger glove.

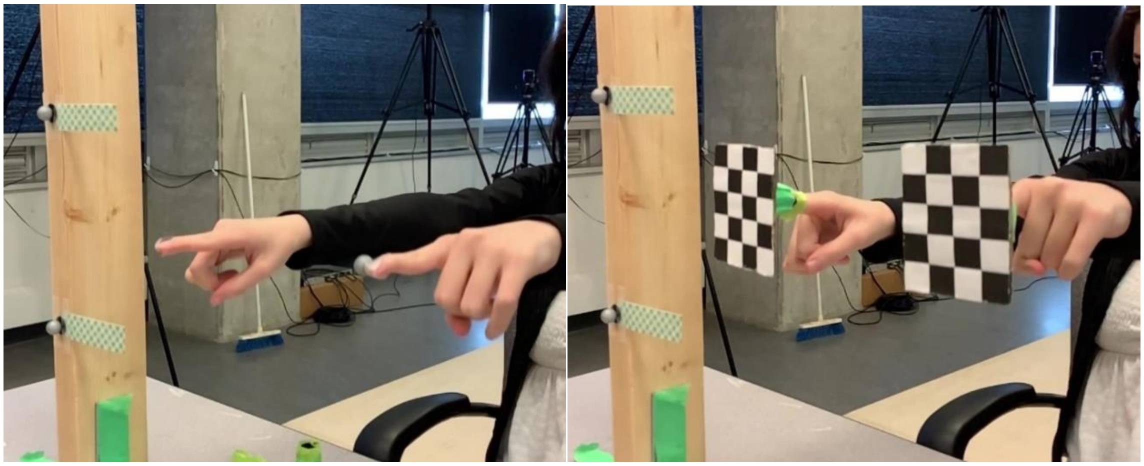


Figure 18: Left: Finger tracking experimental setup. Right: Snapshot of calibration video

Vicon and camera data were collected synchronously. Vicon motion capture was set to a sampling rate of 100 Hz or 200 Hz, depending on the trial. Figure 19 below shows a simplified illustration of the finger tracking data collection apparatus.

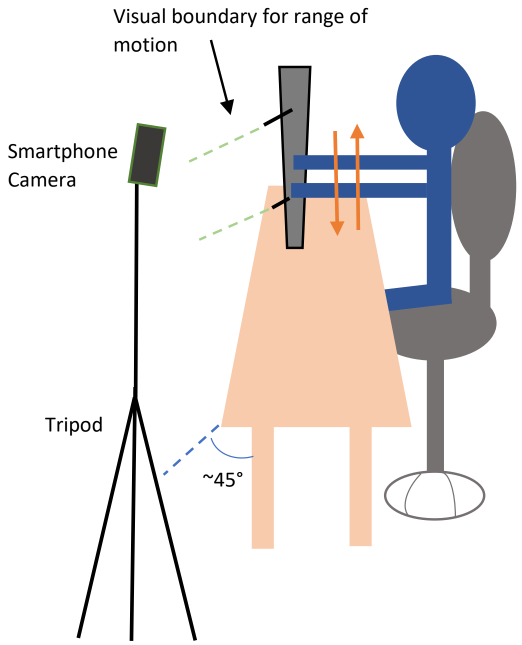


Figure 19: Illustration of experimental set-up

3.4.2 Evaluation Metrics

Bland-Altman Plot

To compare quantitative measurements obtained from two different instruments or methods, the Bland-Altman (BA) plot is typically utilized [52,53]. It provides a visual representation of the agreement between two measurements by plotting their difference against their mean. Firstly, analyzing the BA provides information about the presence of bias in the mean difference of the methods, such that any non-zero mean difference value suggests potential bias in one direction. Second, a BA plot estimates agreement intervals wherein 95% of the difference lies. Finally, looking at the spread of the difference values for different measurements exposes the presence of potential trends in the error.

For all analyses performed using the BA plot in this chapter, the reference method was the Vicon motion capture measurements. As such, the difference values are of the computer vision measurements compared to Vicon's measurements. A positive mean difference would indicate that the Vicon measurements are on average higher than the other method, while a negative mean difference would indicate the opposite bias. The closer the bias value is to zero, the better is the performance of the proposed methods.

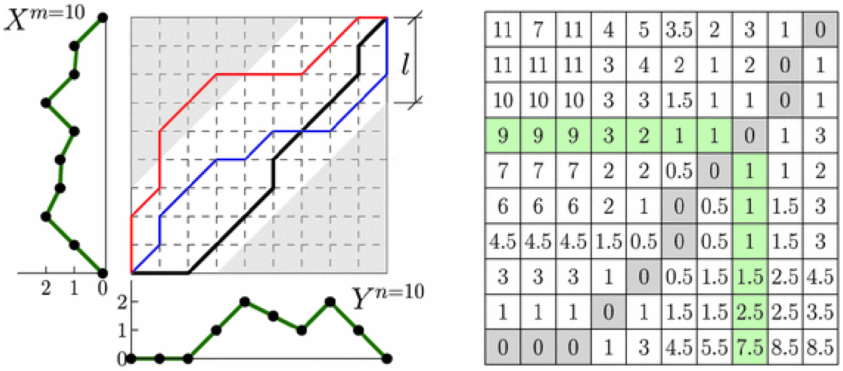
As for the limits of agreement, they are computed using:

$$LoA = mean(A - B) \pm 1.96 * SD(A - B)$$

where A and B are the measurements from the two methods being compared and SD is the standard deviation. Since the mean and standard deviation are computed, the assumption of normality for the difference needs to be valid, which was assessed visually for all trials by plotting the histogram of the difference.

To obtain the BA plot, the BlandAltman MATLAB function was used [54]. Prior to the BA plot, the Vicon data was either downsampled or upsampled to match the vector length of the data being compared to it. The reason Vicon data was interpolated rather than the assessed methods' data was to include every datapoint collected by the assessed method in the analysis. Since Vicon measurements are of high accuracy (error < 1 mm), and the minimum sampling rate was 100 Hz, interpolated data was deemed reliable. Following downsampling, the two data vectors were temporally synchronized using dynamic time warping. Since the two methods have different sampling rates, synchronization was necessary to be able to compare each corresponding datapoint.

Dynamic Time warping (DTW) synchronizes two signals by non-linearly aligning them such that each data point could be matched with many from the other signal, with the use of a cost function. The DTW algorithm computes a distance or cost matrix of the two signals, then selects the smallest path through the matrix from the last point to the first, representing the path for best alignment. Figure 21 depicts aligning signals using Euclidian distance on the top (a), and DTW on the bottom (b) for comparison [55]. Figure 20 shows an example of the distance/cost matrix path selection [56].



(a) Time series alignment.

11	7	11	4	5	3.5	2	3	1	0
11	11	11	3	4	2	1	2	0	1
10	10	10	3	3	1.5	1	1	0	1
9	9	9	3	2	1	1	0	1	3
7	7	7	2	2	0.5	0	1	1	2
6	6	6	2	1	0	0.5	1	1.5	3
4.5	4.5	4.5	1.5	0.5	0	0.5	1	1.5	3
3	3	3	1	0	0.5	1.5	1.5	2.5	4.5
1	1	1	0	1	1.5	1.5	2.5	2.5	3.5
0	0	0	1	3	4.5	5.5	7.5	8.5	8.5

(b) Cumulative distance matrix.

Figure 20: Example of the distance/cost matrix path selection [56]

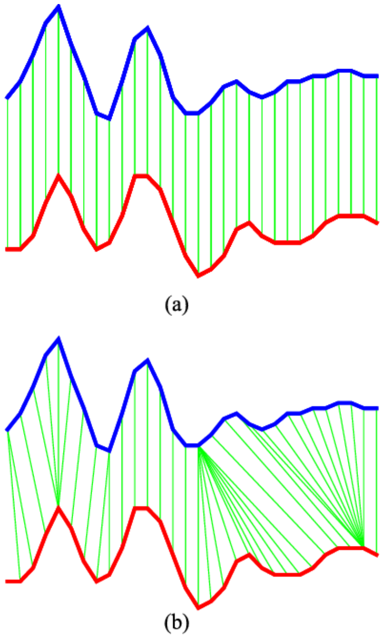


Figure 21: Signals alignment using Euclidian distance on the top (a), and DTW on the bottom (b) [55]

Mean Peak-to-peak Distance and Peak Velocity Errors

Since two of the most significant time-domain features derived in Chapter 2 were the mean pk-pk distance and the mean peak velocity, they were also used as metrics for assessing the accuracy of the methods in this section. Pk-pk distance was computed by finding the distance between each peak and valley then comparing each pk-pk distance from the Vicon data to that of the computer vision data. Peak velocity was obtained by first finding the velocity through differentiating the distance, then finding the peaks and valleys. These were also individually compared for each peak. The mean of the error/difference between the pk-pk and peak velocities of the two methods is also computed.

Required Effect Size

A one-way ANOVA was performed to obtain an effect size, from which the accuracy constraint was obtained. The ANOVA was performed for the mean pk-pk distance and mean peak velocity data of concussed and healthy participants, using the experimental design described in Chapter 2. Normality of residuals and homogeneity of variance were assessed to ensure ANOVA assumptions were not violated.

The results showed that both these features were statistically significant with p-values of 0.0169 and 0.0105 for the mean amplitude and velocity, respectively. Ad-hoc analysis was performed to obtain the difference between the group means, as these differences will be used as the effect sizes to assess the accuracy of the proposed collection methods. For the mean amplitude, the difference between group means was 37.97 mm. As for the mean velocity, the difference between group means was 131.74 mm/s. These were the values used to set the constraints listed in the previous section. Figure 22 below visually illustrates the difference. Note that this is the maximum allowable error for a method to be considered; however, lower error values are required in practice for higher sensitivity.

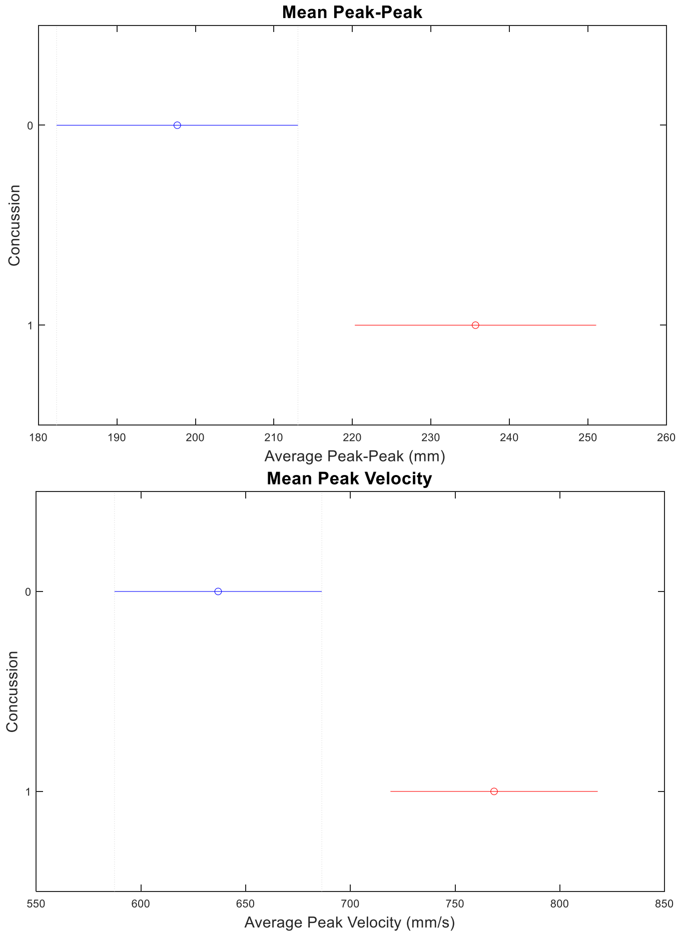


Figure 22: Top: Difference in group means between the concussed and healthy groups for the mean peak to peak distance. Bottom: Difference in group means between the concussed and healthy groups for the mean peak velocity

Calibration Curve

Since the vertical pixel distance changes throughout the trial as the participant moves their hands, the calibration ratio is unlikely to be constant. This is mainly attributed to the video perspective, such that the pixel distance will be higher at the top of the image frame than at the bottom, for the same real world distance. The camera tilt was intentionally introduced as part of the experimental settings for more consistent hand detection. Other changes in pixel distances are caused by the natural spheroidal movement of the arm, which not only impacts the pixel distance but also the real-world distance used. This is because the real-world reference distance (square length) that is being used for calibration assumes that this reference distance is always equal to the height of the square; however, that will not be true at the extremes of the motion where the actual height would be the vertical component of the square length, due to the angle introduced by the shoulder joint. In addition, there might be small changes also attributed to lens distortions. Figures 23 illustrates the issue, wherein snapshots of the video were taken at a peak and a valley of the movements. This issue is more prominent in trials where the camera is more angled.

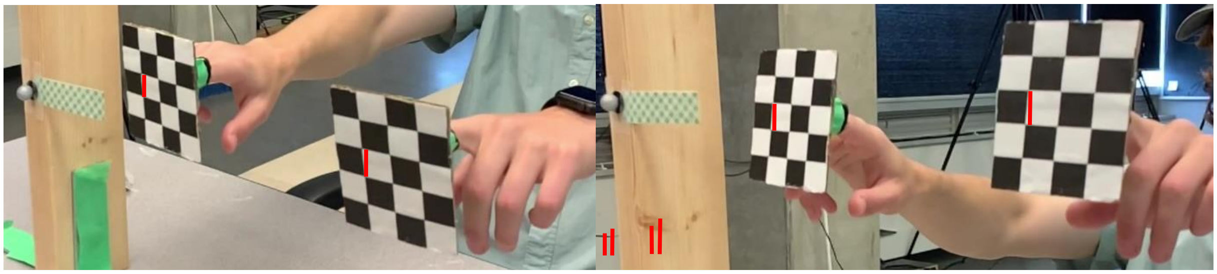


Figure 23: Snapshots of the calibration video at the top and bottom of the calibration range, showing the change in vertical pixel distance depending on location

While the trial shown in Figure 23 above showed only small differences, accounting for this discrepancy may improve accuracy, and was pursued to examine potential accuracy improvements.

To account for the difference in pixel distances caused by the camera tilt/perspective, a calibration curve was obtained rather than a constant. To obtain the calibration curve, steps 1-4 from Algorithm 2 were followed, then the steps described in Algorithm 3 were performed.

Algorithm 3 Calibration Curve

```
Input: Calibration video
Output: Calibration curve
Signal← Mediapipe trial data for the hand of interest
Signal=-Signal
Filter signal using 3rd order Butterworth low pass filter with a $F_c$=15 Hz
Apply Hampel filter to remove outliers
Rescale signal to be in range (-1, 1)
Set fraction of the peaks' surrounding points to 1, and of the valleys' surrounding points to -1
Set a fraction of surrounding points of the new extremes to nan
Fill missing values using shape-preserving piecewise cubic spline interpolation
Apply median filter
Rescale signal←max=pixel distance peak, min= pixel distance valley
Calibration curve=Real square height/ rescaled signal
Rescale calibration curve so that the valleys = mean (max, min)
```

Considering real world data for this application is unlikely to behave with perfect stationarity, rescaling the data to obtain the calibration curve was done to match the frequency of the data and to match the unique shape of the peaks.

Figure 24 below shows an example of a calibration curve, and Figure 25 shows the Mediapipe data for one hand along with its magnified calibration curve, to illustrate how the data will be multiplied by its calibration curve. It should be noted that the Mediapipe data was vertically flipped such that a larger value indicates the upward direction in the image frame.

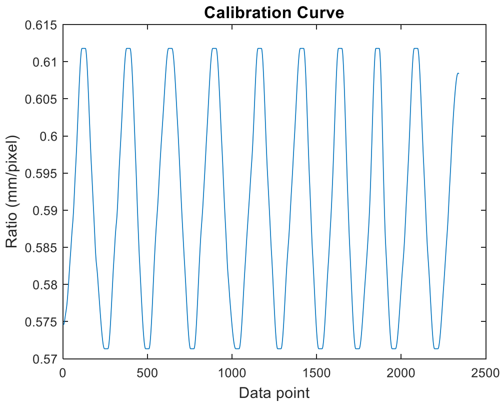


Figure 24: Example of calibration curve for one of the trials

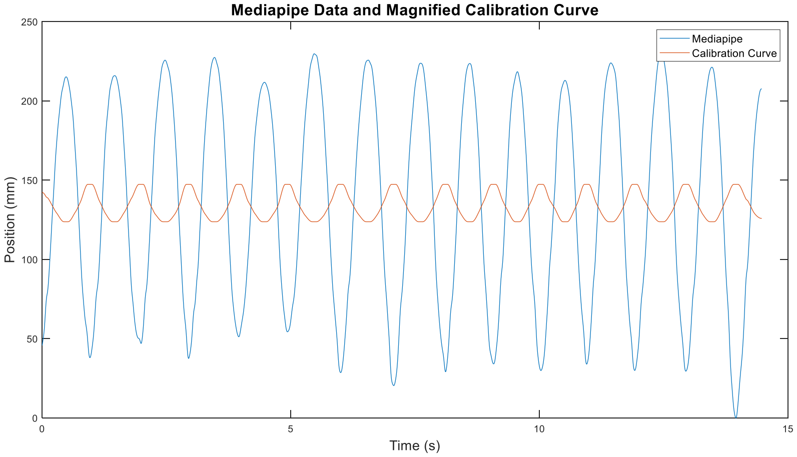


Figure 25: Mediapipe data for one trial and corresponding magnified calibration curve to illustrate how the curve is multiplied by the data

The graphical results obtained using the calibration curve method are shown in Appendix A.

3.5 Results

The results obtained from two different participants are plotted in Figures 26 to 42 for each of the four methods below. Selected plots are shown for each trial, the rest of the plots are in Appendix A. One of the plots shows the synchronized position curves of the two methods, using dynamic time warping. The second plot type shown is the Bland Altman, to illustrate the difference between the two measurements for each data point collected by the proposed method. Finally, the last plot shows the error in peak to peak distance for each peak, and the error in the peak velocity for each peak. To obtain the synchronized and peak error plots, the two signals were truncated to begin with the first peak and end in the last peak. In all the cases, the Vicon data was filtered using a 3rd ordered Butterworth filter with a cut-off frequency of 15 Hz.

Colour Tracking

In the colour tracking evaluation, the mean bias was 0.008 and 0.315 mm for the left and right hand, respectively. Average limits of agreement, as shown in the B-A plots were -7.2 to 7.2 mm (left hand) and -8.6 to 9 mm (right hand). Errors in pk-pk distance and peak velocity were 14.8 mm and 45.75 mm/s respectively. Figure 26 below shows the colour masked image. The two white circles are the yellow markers, which were tracked in each frame to obtain the following results. The data was filtered using a 3rd ordered Butterworth filter with a cut-off frequency of 15 Hz, followed by Hampel filter for removing potential outliers. The results for both hands are shown graphically below in Figures 27 to 30 and in Appendix A.

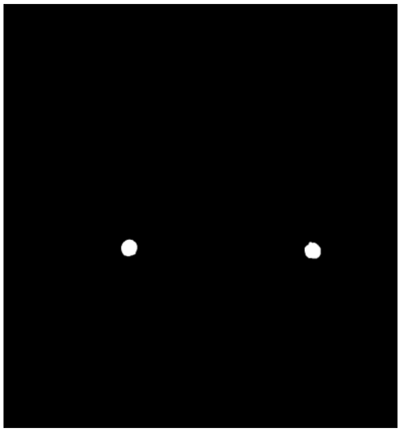


Figure 26: Colour masked image for yellow marker tracking

P1: Left Hand

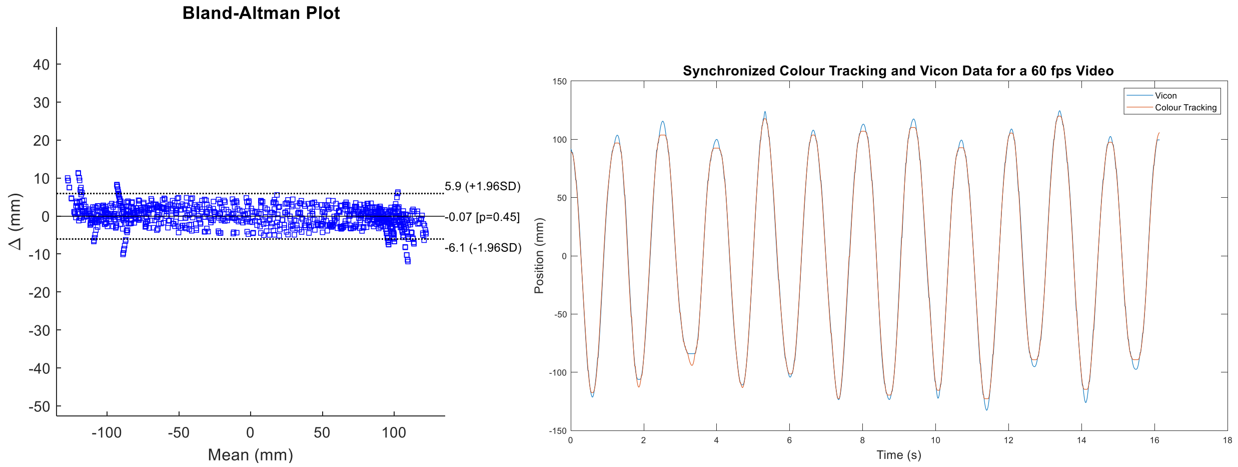


Figure 27: Left: Bland Altman plot comparing position data obtained from colour tracking and Vicon motion capture. Right: Synchronized colour tracking and Vicon position data using DTW

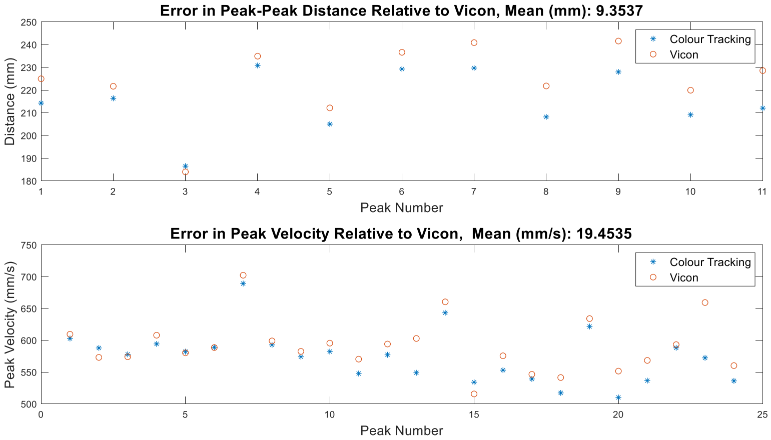


Figure 28: Top: Error in peak to peak distance between colour tracking and Vicon. Bottom: Error in peak velocity between the two methods

As can be seen from Figure 28, the error is higher at the extremes where the peaks and valleys occur. This points to poorer tracking at the top and bottom of the motion attributed to poorer calibration as the marker deviates away from the image center. The graphical results for participant 2, right hand are shown below. The results for the left hand are shown in Appendix A.

P2: Right Hand

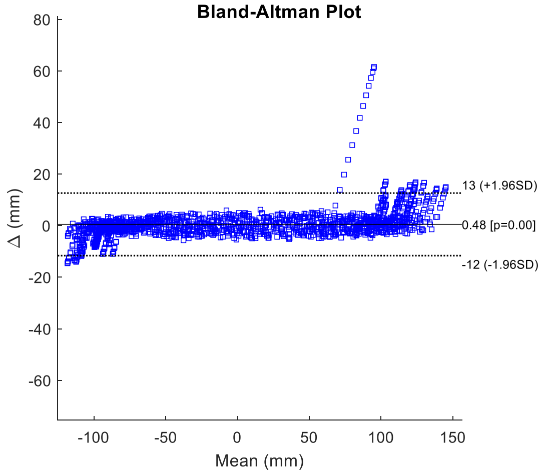


Figure 29: Bland Altman plot comparing position data obtained from colour tracking and Vicon motion capture

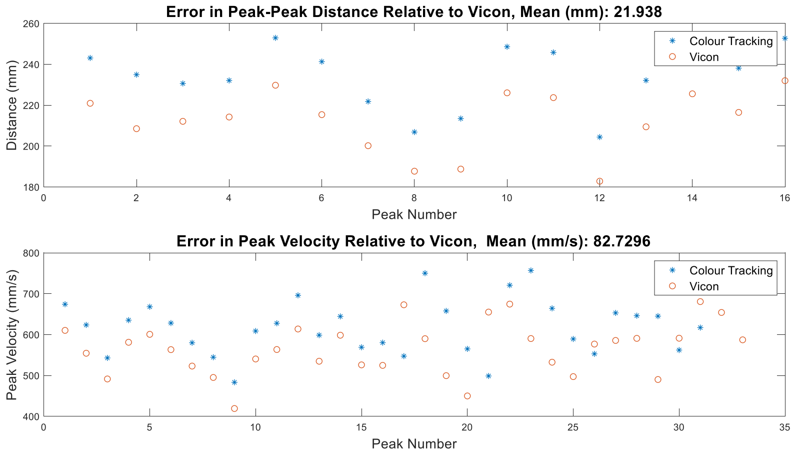


Figure 30: Top: Error in peak to peak distance between colour tracking and Vicon. Bottom: Error in peak velocity between the two methods

Bounding Box

The output of Handtrack.JS consists of the length and width of the bounding box in pixels as well as the pixel coordinates of the top left corner of the bounding box. To obtain the pixel position of the bounding box’s centroid with respect to the whole image, the following was done:

$$Center2 = y + \frac{Length_{BB}}{2}$$

where Center 2 is the centroid of the bounding box with respect to the whole image, which is the desired output, y is the pixel coordinate of the corner of the bounding box (in the y direction), and length_{BB} is the length of the bounding box. Following that, the data was filtered using a 3rd order Butterworth filter with a cut-off frequency of 15 Hz, followed by Hampel filter for removing potential outliers.

From the output using the bounding box hand tracking method, the mean bias was 0.625 and 0.745 mm for the left and right hand, respectively. Average limits of agreement, as shown in the B-A plots were -10.25 to 9.55 mm (left hand) and -10.75 to 9.35 mm (right hand). Errors in pk-pk distance and peak velocity were 48.8 mm and 355.5 mm/s respectively. The results for both hands are shown graphically below in Figures 32 to 34, note that only selected figures are shown here, others can be found in Appendix A. Figure 31 below shows the bounding box detected for one of the frames, along with the prediction confidence.

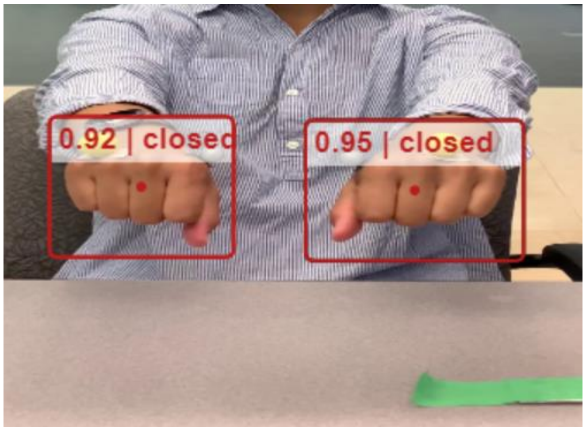


Figure 31: Bounding box detected using Handtrack, with the prediction confidence

P1: Right Hand

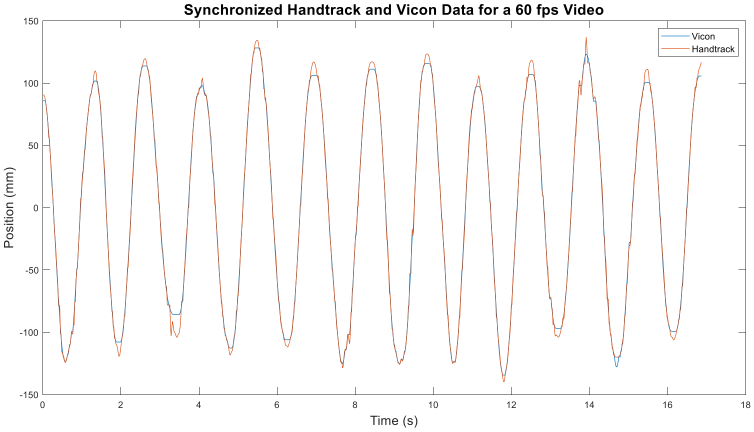


Figure 32: Synchronized Handtrack and Vicon position data using DTW

As can be seen from Figure 32 above, there are some noisy regions, mainly in the peaks and valleys, pointing to poor tracking at the extremes. However, in the case of the bounding box, this is likely attributed to misclassifications rather than calibration inaccuracies.

P2: Right Hand

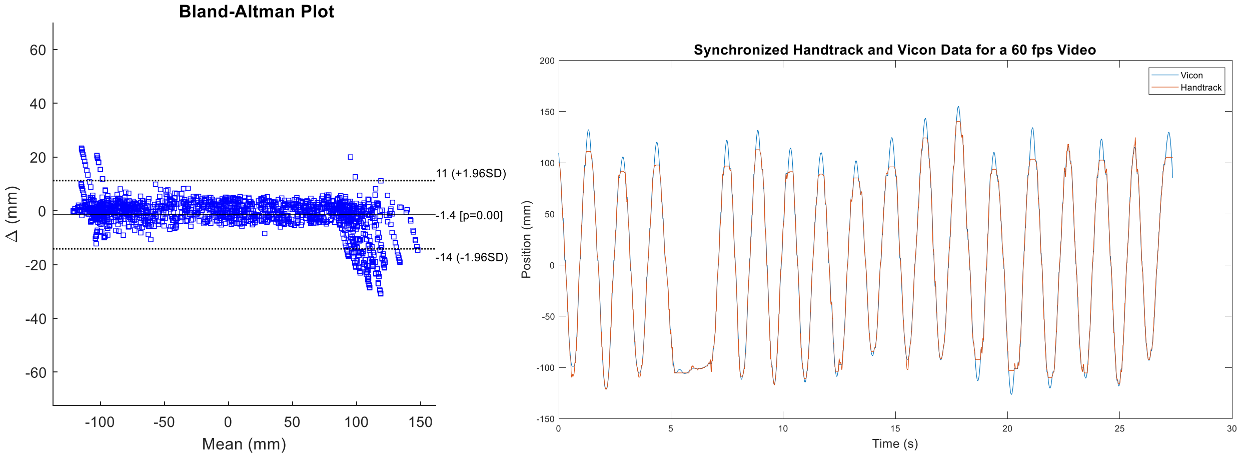


Figure 33: Left: Bland Altman plot comparing position data obtained from Handtrack and Vicon motion capture. Right: Synchronized Handtrack and Vicon position data using DTW

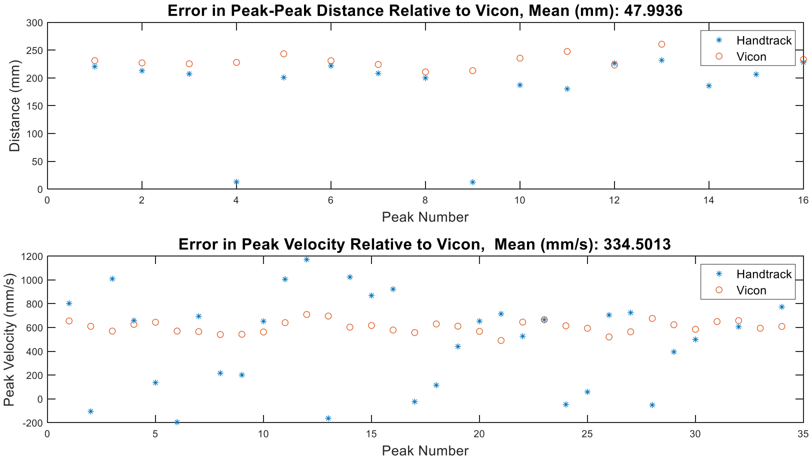


Figure 34: Top: Error in peak to peak distance between Handtrack and Vicon. Bottom: Error in peak velocity between the two methods

Similar to participant 1, participant 2 also exhibited higher tracking errors at the extremes. In this case however, the error is higher for peaks (compare to valleys). Furthermore, rather than noise at the peaks, it seems to be tracking an incorrect target at the top of the image.

Hand Fist Skeleton

To process the data and obtain the landmarks of interest from the Mediapipe output, only frames with two detected hands were kept. Frames wherein only one hand or zero hands were detected were substituted by NaN. Following that, the y-axis pixel positions for landmarks 5 and 8 were extracted and stored, for each hand. NaN values were then replaced with values from spline interpolation. Mediapipe provides landmark locations that are normalized to the image height and width; therefore, the y-direction pixel positions were then multiplied by the image height. Finally, the data was filtered using a 3rd order Butterworth filter with a cut-off frequency of 15 Hz, followed by Hampel filter for removing potential outliers. For the fist tracking described in this section, only landmark 5 was used.

From the output of the landmark tracking method, the mean bias was 0.525 and 0.635 mm for the left and right hand, respectively. Average limits of agreement, as shown in the B-A plots were -9.25 to 10.01 mm (left hand) and -10.1 to 11.35 mm (right hand). Errors in pk-pk distance and peak velocity were 40.6 mm and 367.6 mm/s respectively. The results for both hands are shown graphically below in Figures 36 to 37, and in Appendix A. Figure 35 shows the landmarks detected in a frame with the fist position.

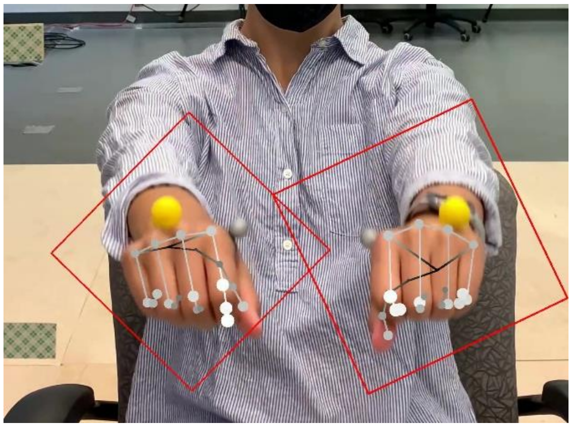


Figure 35: Hand landmarks detected for hand fist using MediaPipe

P1: Left Hand

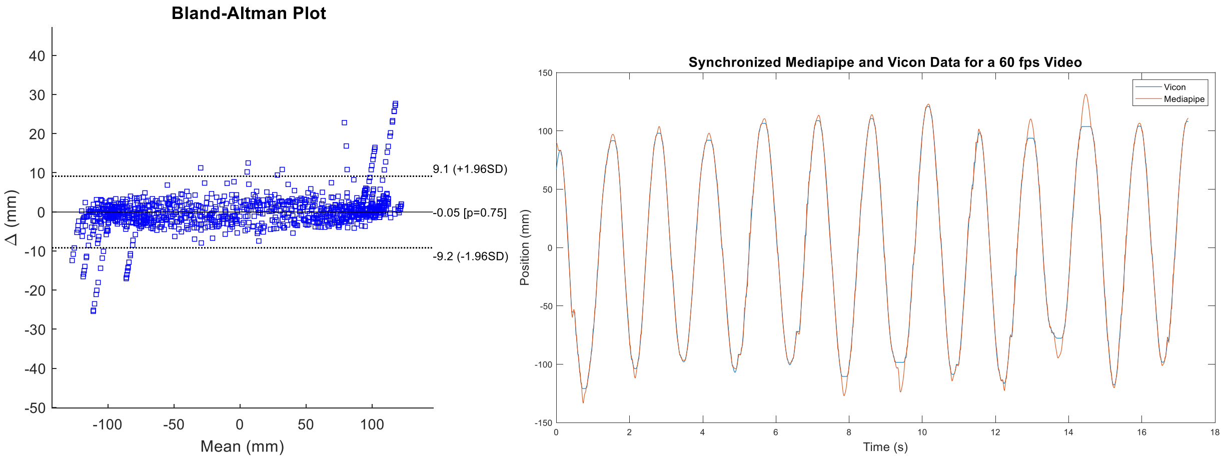


Figure 36: Left: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture. Right: Synchronized Mediapipe and Vicon position data using DTW

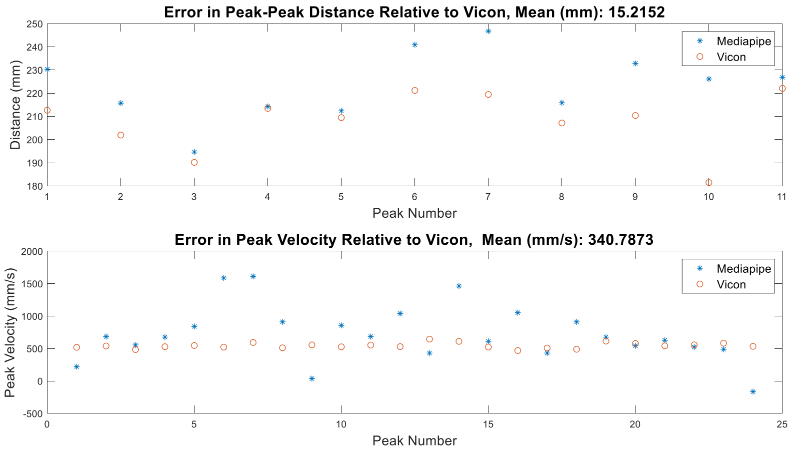


Figure 37: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between the two methods

As can be seen in Figure 37 above, the valleys are poorly tracked due to poor landmark detection when the hand is near the bottom of the image frame and likely attributed to poorer landmark detection for that hand perspective (birds eye view of the hand). Additionally, motion blur leading to incorrect knuckle localization contributed to poor tracking.

Finger Tracking

The same processing procedure described in the previous section (fist landmark tracking) was used for finger landmark tracking, except landmark 8 was extracted instead. Figure 38 below shows the landmarks detected in a frame with the finger tracking configuration. Note that the pictures are cropped to focus on the hands. This method was able to detect the hand skeleton fairly well despite a very noisy background.

From the output of the landmark tracking method, the mean bias was 0.2 and 0.7 mm for the left and right hand, respectively. Average limits of agreement, as shown in the B-A plots were -16.0 to 17.1 mm (right hand) and -9.9 to 10.2 mm (left hand). Errors in pk-pk distance and peak velocity were 21.19 mm and 279.69 mm/s respectively. Figure 38 on the right illustrates the issue of motion blur, which degrades the tracking performance, which was an issue for all the 60 fps videos. The results for both hands are shown graphically below in Figures 39 to 42 and in Appendix A.

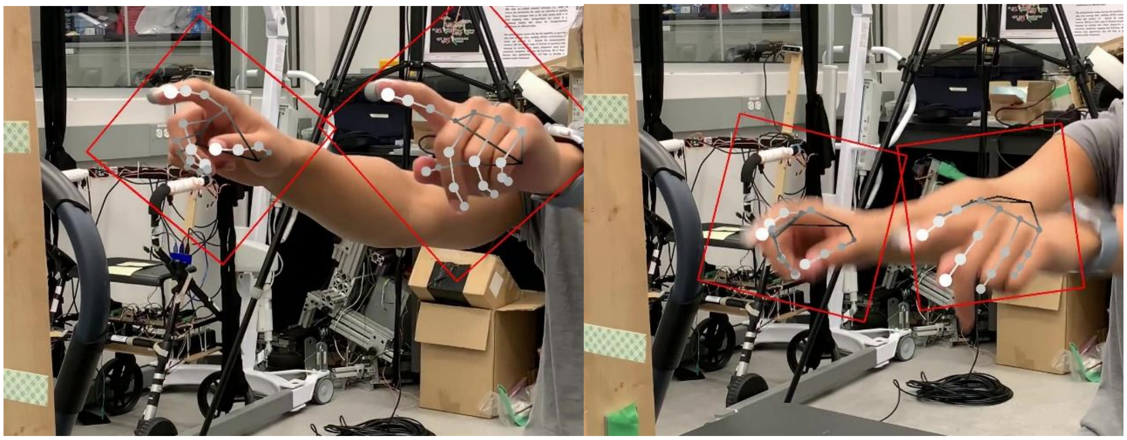


Figure 38: Left: Hand landmarks used for finger tracking using Mediapipe. Right: Effect of motion blur on detection performance

P1: Right Hand

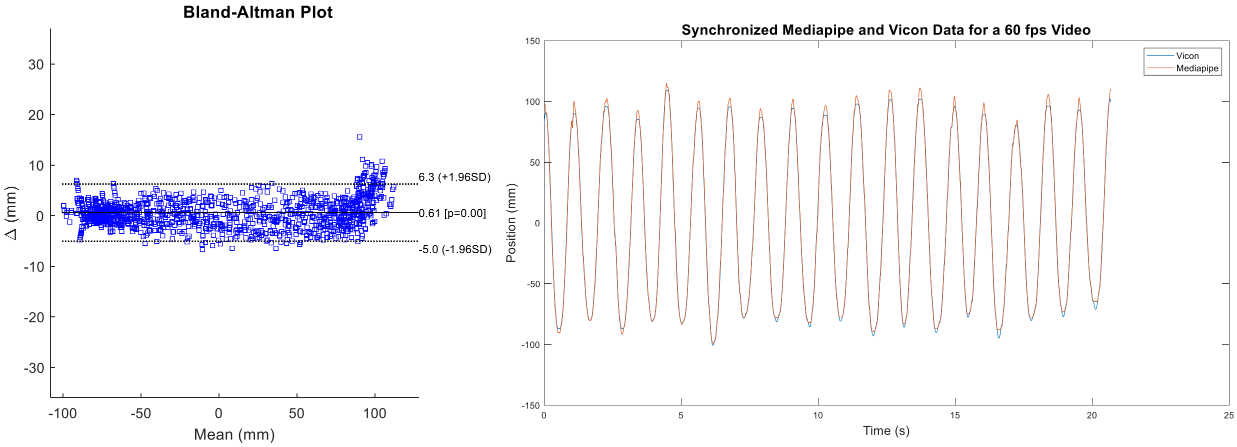


Figure 39: Left: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture. Right: Synchronized Mediapipe and Vicon position data using DTW

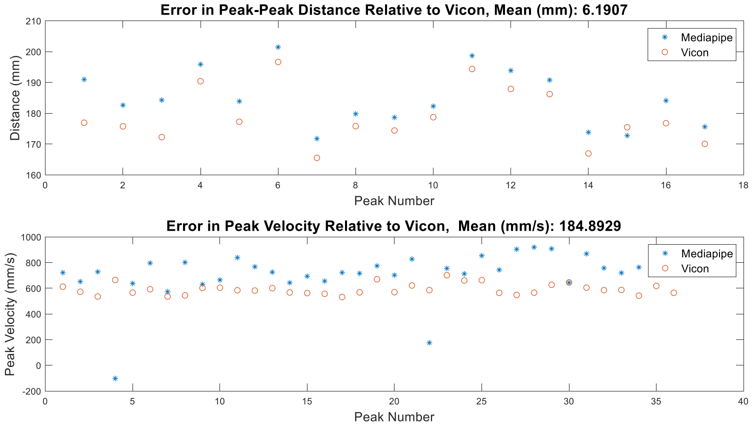


Figure 40: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between the two methods

P2: Right Hand

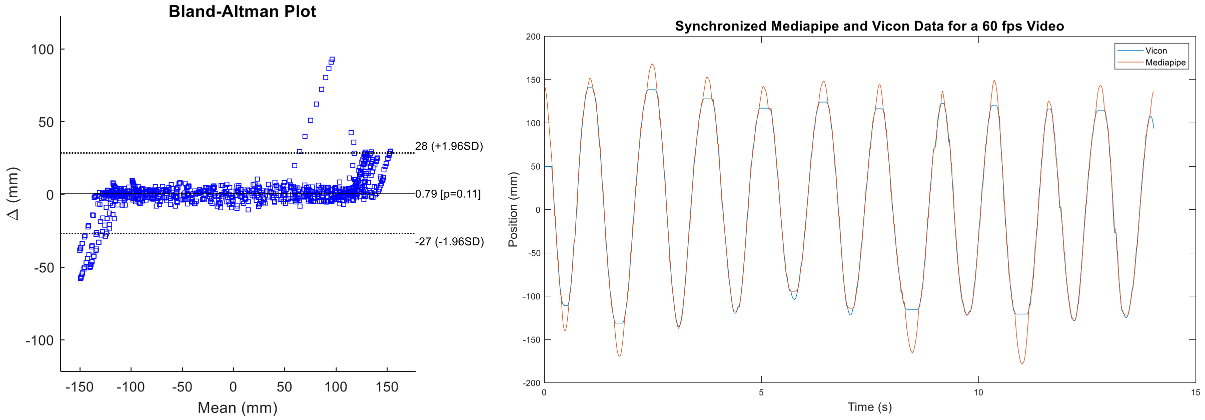


Figure 41: Left: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture. Right: Synchronized Mediapipe and Vicon position data using DTW

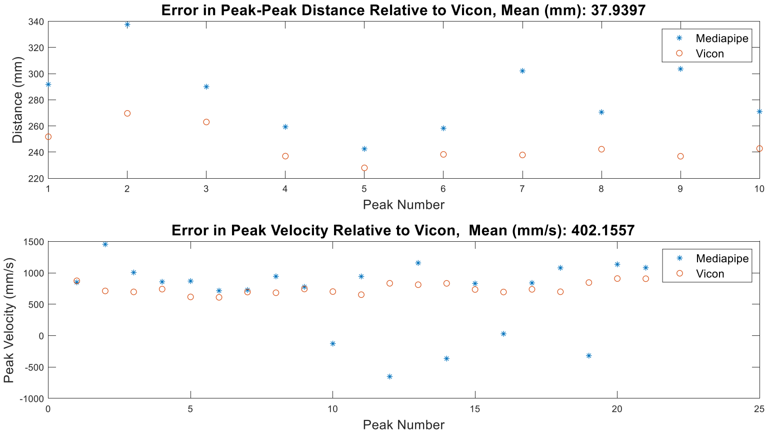


Figure 42: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between the two methods

The figures above illustrate the difference in the tracking performance between participants 1 and 2. This emphasizes the importance of the collection settings (i.e., angle, tilt, distance away from camera, etc.) on performance.

Comparison

Numerical results for the trials are tabulated in Table 17 below. The Bias and LoA represent metrics obtained from the Bland-Altman plot shown in the graphical results above and in Appendix A. LoA is the limits of agreement between the two methods, wherein the first value is the lower bound and the second is the upper bound. A negative bias value means that the Vicon distances were on average lower than the distances obtained through the method being studied, while a positive value points to the opposite trend. Peak-peak error and peak velocity error depict the mean error for all the peaks in the trial and are also shown graphically in the previous section and in Appendix A.

Table 17: Summary of results obtained for the different hand tracking methods

Method			Bias (mm)	LoA (mm)	Pk-pk Error mean (SD) (mm)	Pk Vel. Error mean (SD) (mm/s)
Colour tracking	***P1***	***R***	-0.15	-5.2, 5.0	7.10 (3.32)	30.05 (15.47)
		L	-0.07	-6.1, 5.9	9.35 (4.41)	19.45 (19.07)
	P2	***R***	0.48	-12.0, 13.0	21.93 (9.90)	82.72 (45.18)
		L	0.10	-8.3, 8.5	18.91 (12.60)	52.22 (40.25)
Bounding box	***P1***	***R***	0.09	-7.5, 7.7	34.82 (47.38)	432.83 (292.78)
		L	0.29	-6.5, 7.1	43.52 (73.07)	315.30 (242.44)
	P2	***R***	1.40	-14.0, 11.0	47.99 (65.47)	334.50 (233.07)
		L	-0.96	-14.0, 12.0	68.89 (62.81)	339.41 (266.07)
Fist Landmarks	***P1***	***R***	0.77	-8.2, 9.7	36.80 (49.33)	232.97 (308.77)
		L	-0.05	-9.2, 9.1	15.21 (13.10)	340.78 (312.20)
	P2	***R***	0.50	-12.0, 13.0	34.60 (46.96)	446.23 (383.74)
		L	1.00	-9.3, 11.0	75.85 (60.61)	486.61 (389.33)
Finger Tracking	***P1***	***R***	0.61	-5.0, 6.3	6.19 (2.90)	184.89 (140.53)
		L	0.29	-4.8, 5.4	5.18 (2.00)	208.73 (190.23)
	P2	***R***	0.79	-27.0, 28.0	37.93 (20.76)	402.93 (426.90)
		L	0.12	-15.0, 15.0	35.48 (45.77)	310.21 (312.70)

While all methods for the fist tracking show relatively low bias and acceptable LoA, the peak to peak error and peak velocity error for most trials do not meet the constrained maximum allowable errors of 37 mm and 131 mm/s. Colour tracking is the only method that meets the accuracy constraints. The fist landmark method met the peak to peak error requirement for most trials except for participant 2’s left hand. The bounding box did not meet the peak to peak error for majority of the trials. As for the peak velocity, the markerless methods exceeded the maximum allowable error, likely due to noise in the data. Note that the selected method will mainly focus on improving the pk-pk error, as well as having reasonable bias and LoA ranges. In the case that the peak velocity estimate is the only poorly estimated metric, it can be replaced by another informative metric which is better estimated using the computer vision method. This is because the velocity estimate relies on computing the derivative which amplify noise, while other features computed in Chapter 2, such as the frequency-domain features, are less prone to noisy data. The finger tracking (with 60 fps video input) has the potential to yield acceptable results; however it seems to be dependent on the trial conditions. The peak to peak error values fall within the acceptable range, except for participant 2's right hand data, wherein it's ~0.93 mm higher. In addition, it results in a noisier output, likely attributed to motion blur and the lower number of data points collected, which results in higher peak velocities, outside of the acceptable range for this task.

3.6 Modifications to Selected Method

The method selected for further analysis was the finger landmark tracking. Firstly, it more closely resembles the settings of the original experiment. Second, the results look promising for participant 1, which shows that given proper experiment execution, this method has the potential to produce the desired results. Furthermore, given other design requirements, markerless tracking is superior to the coloured marker tracking, and the finger tracking is the best performing markerless tracking method.

To investigate further, two areas were further modified: the video frame rate, and the calibration procedure. Due to the camera's resolution to framerate trade-off, the 60 fps video has a higher resolution of 4k while the 120 fps and 240 fps videos have a resolution of 1K. We hypothesized that the higher framerate with lower resolution will improve results compared to lower framerate with higher resolution. The reason is that at 60 fps, motion blur is evident and likely plays a role in degrading accuracy of the landmark detection. Moreover, lower framerates also affect calibration as even a small number of unaccounted pixels could contribute to accuracy offsets.

Frame Rate

Two frame rates were tested: 120 fps and 240 fps. Both of these framerates more closely resembled the sampling rates for the motion capture system which were either 100 or 200 Hz. Figure 43 illustrate examples of the finger tracking results for these framerates, mainly to show the lack of motion blur in the image snapshots from the video.

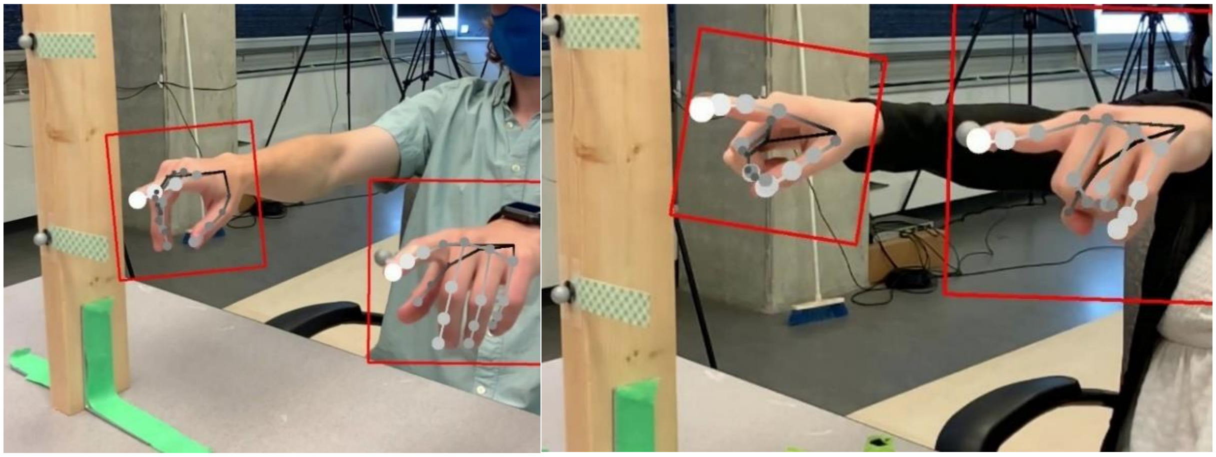


Figure 43: Examples of hand landmarks detected from a slow-motion video, using Mediapipe

The results for each of the framerates are shown below, in Figures 44 to 59, also for 2 different participants.

Results – Graphical

From the output of the 120 fps video landmark tracking method, the mean bias was 0.28 and 0.2 mm for the left and right hand, respectively. Average limits of agreement, as shown in the B-A plots were -3.2 to 3.6 mm (right hand) and -4.1 to 4.1 mm (left hand). Errors in pk-pk distance and peak velocity were 6.96 mm and 102.2 mm/s respectively.

<u>120 fps - *P1:* Left Hand</u>

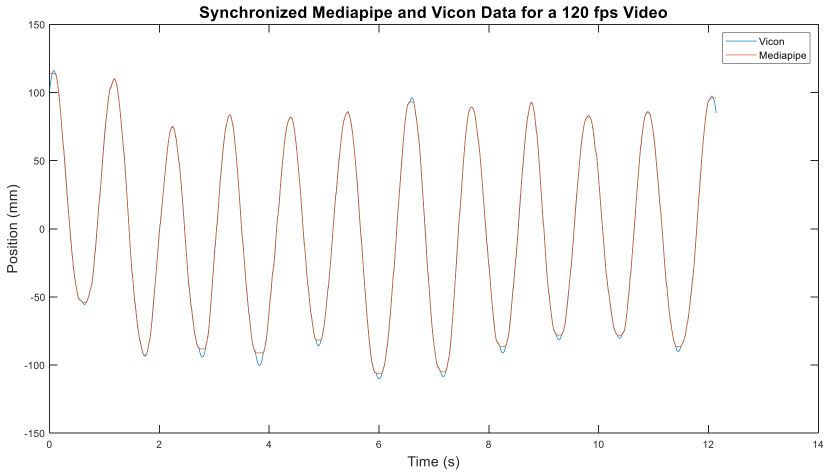


Figure 44: Synchronized Mediapipe and Vicon position data using DTW for a 120 fps video

P2: Right Hand

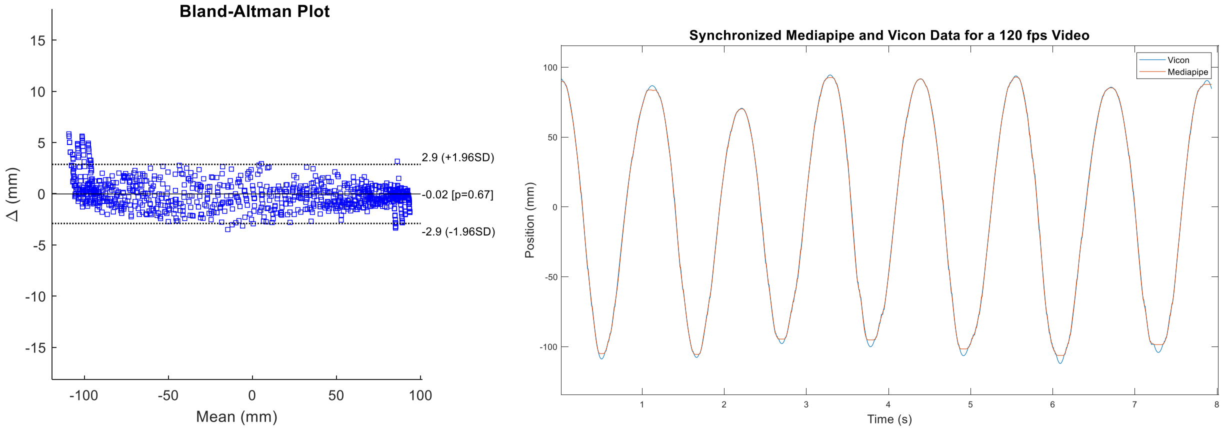


Figure 45: Left: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 120 fps video. Right: Synchronized Mediapipe and Vicon position data using DTW for a 120 fps video

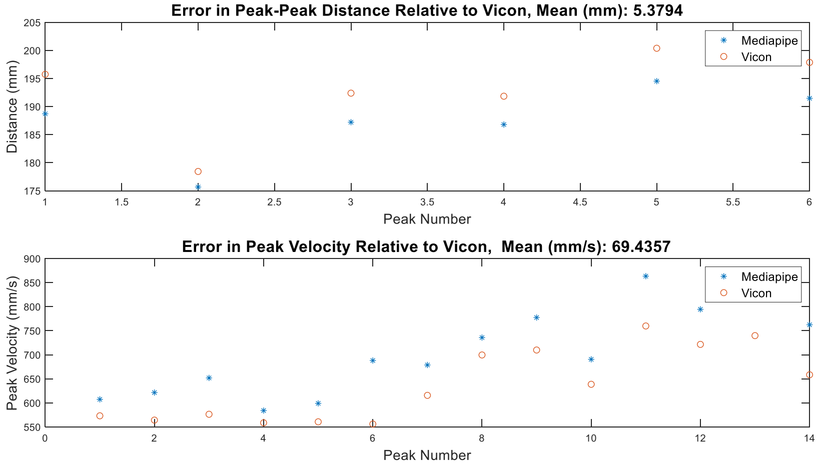


Figure 46: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between the two methods

For the output of the 240 fps video landmark tracking method, the mean bias was 0.055 and 0.085 mm for the left and right hand, respectively. Average limits of agreement, as shown in the B-A plots were -2.3 to 2.75 mm (right hand) and -2.65 to 2.6 mm (left hand). Errors in pk-pk distance and peak velocity were 4.89 mm and 101.6 mm/s respectively.

240 fps - *P1:* Right hand

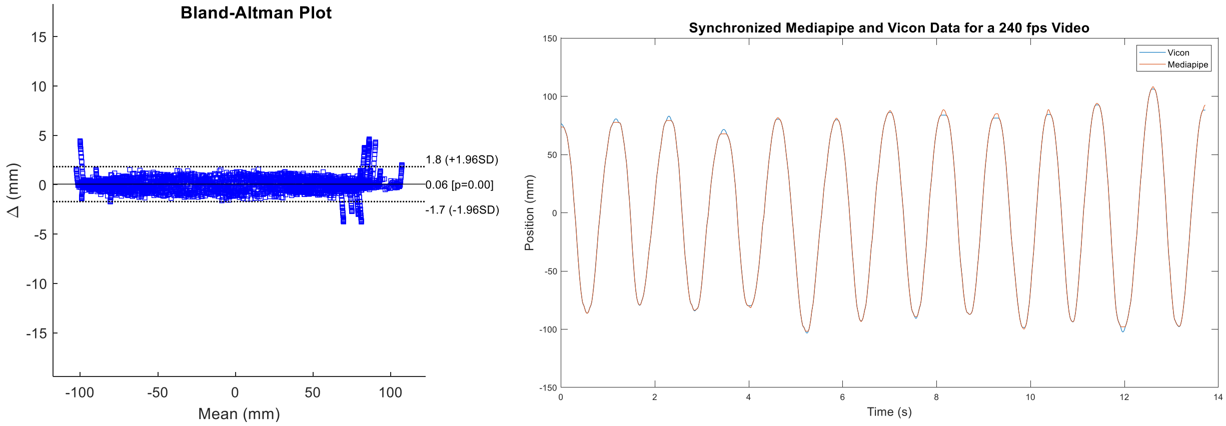


Figure 47: Left: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 240 fps video. Right: Synchronized Mediapipe and Vicon position data using DTW for a 240 fps video

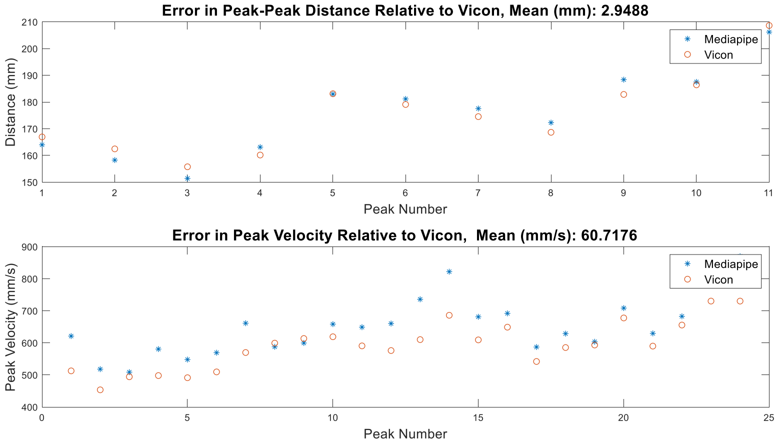


Figure 48: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between the two methods

<u>*P2:* Right hand</u>

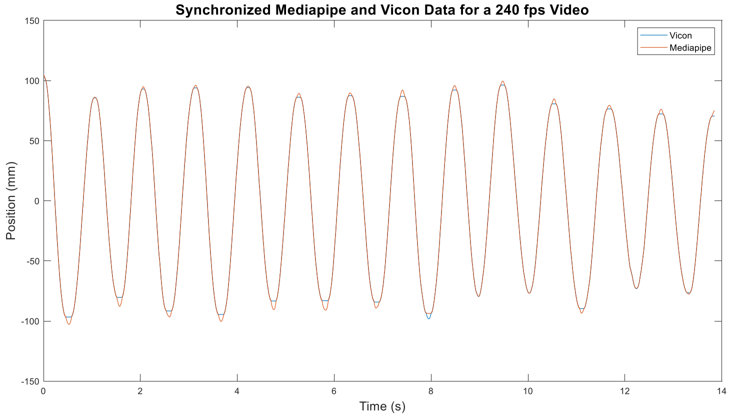


Figure 49: Synchronized Mediapipe and Vicon position data using DTW for a 240 fps video

Results-Modifications

The numerical results for finger tracking with higher framerates are tabulated below. The same metrics are used as in the previous tracking analysis, and the same interpretations apply. Table 18 contains the results wherein a calibration constant was used for position data calibration, whereas table 19 contains the results where a calibration curve was used instead. There are two main comparisons to note: The effect of frame rate on performance, and the type of calibration procedure that yields improved performance.

Table 18: Summary of results for finger tracking using MediaPipe with higher video frame rates, and a calibration constant

fps			Bias (mm)	LoA (mm)	Pk-pk Error mean (STD) (mm)	Pk Vel. Error mean (STD) (mm/s)
120	***P1***	***R***	0.38	-3.5, 4.3	6.88 (2.94)	97.58 (84.93)
		L	0.31	-2.8, 3.4	3.89 (2.88)	42.81 (34.31)
	P2	***R***	-0.02	-2.9, 2.9	5.37 (1.47)	69.43 (32.51)
		L	-0.25	-5.4, 4.9	11.73 (2.86)	199.64 (56.80)
240	***P1***	***R***	0.06	-1.7, 1.8	2.94 (1.51)	60.71 (38.25)
		L	0.08	-2.7, 2.5	6.41 (2.79)	123.69 (38.65)
	P2	***R***	0.11	-2.9, 2.7	6.60 (3.14)	101.43 (51.66)
		L	0.03	-2.6, 2.7	3.64 (2.61)	120.90 (180.25)

From Table 18 above, the overall accuracy improved considerably with increasing framerate. In addition, the results were observed to be more consistent and less dependent on the specific trial as the frame rate is increased.

For the 120 fps video, the results are significantly improved. All the peak to peak errors are much lower than the maximum acceptable error value of 37 mm. The error in peak velocities are all within the maximum acceptable error value of 131 mm/s, except for participant 2’s left hand data which resulted in a mean peak velocity error of 199.64 mm/s. Following further analysis of the trial, this error is likely attributed to the calibration procedure. Other than this value, the results of the 120 fps video seem consistent among the trials and yield acceptable performance as per the needs of this task.

Finally, the 240 fps video yielded acceptable overall performance and met all the error constraints set for this problem.

Table 19: Summary of results for finger tracking using MediaPipe with different video frame rates, and a calibration curve

fps			Bias (mm)	LoA (mm)	Pk-pk Error mean (STD) (mm)	Pk Vel. Error mean (STD) (mm/s)
60	***P1***	***R***	0.57	-4.7, 5.9	5.87 (2.96)	176.28 (138.44)
		L	0.02	-5.0, 5.0	4.90 (2.23)	175.90 (168.37)
	P2	***R***	0.31	-27.0, 28.0	34.88 (22.69)	391.86 (418.23)
		L	0.17	-15.0, 15.0	35.77 (46.19)	345.66 (341.38)
120	***P1***	***R***	0.32	-3.5, 4.2	7.13 (2.91)	95.18 (84.10)
		L	0.2	-2.8, 3.2	4.31 (2.68)	40.88 (33.59)
	P2	***R***	-0.04	-2.8, 2.7	5.34 (1.45)	72.56 (33.44)
		L	-0.61	-6.2, 5.0	11.78 (2.64)	217.42 (58.54)
240	***P1***	***R***	-0.04	-1.8, 1.7	2.95 (1.67)	64.71 (38.65)
		L	-0.2	-2.9, 2.5	5.67 (2.51)	118.97 (39.26)
	P2	***R***	-0.2	-3.2, 2.8	6.37 (3.26)	101.72 (53.37)
		L	-0.05	-2.7, 2.6	3.42 (2.73)	122.60 (187.58)

The effect of using a calibration curve on the results was inconsistent across trials. It improved performance in some cases, deteriorated in others, and yielded no difference for some trials. This is likely attributed to the collection procedure and will be discussed further in the following section.

Table 20 in Appendix B highlights some common data collection errors that lead to deteriorated performance, and should be avoided during the collection procedure. Figure 50 shows examples of good collection settings yielding good performance.

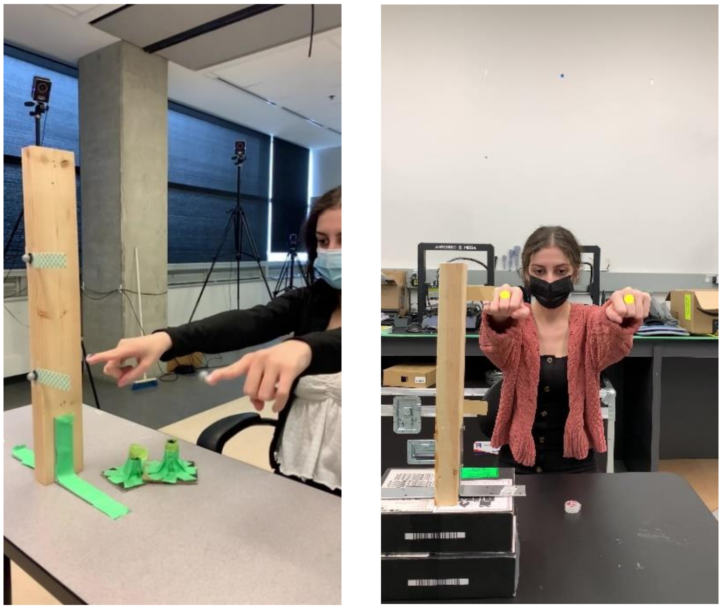


Figure 50: Uncropped images showing good camera and participant arrangement/orientation for collection for a) finger tracking b) fist tracking

3.7 Discussion and Current Limitations

The goal of this chapter was to design and evaluate a portable data collection tool for obtaining position data of upper limbs during the bimanual coordination task (proposed in Chapter 2). Computer vision methods were implemented and tested for hand and finger tracking. Initially, hand tracking was performed by tracking the hand fist. As a less complex alternative to finger tracking and to characterize baseline performance, hand fist tracking was assumed an acceptable alternative to acquire high-quality data obtained through the selected computer vision method, rather than the original motion capture data (finger tracking). As such, fist tracking was considered an acceptable replacement as the same patterns would be produced. Following fist tracking, finger tracking was examined using more complex alternatives, which yielded significantly improved results.

The first hand fist tracking method, colour tracking, yielded good accuracy. However, it failed to meet several other of the outlined constraints mainly because it is not robust and often required slight adjustments to the mask and filtering parameters depending on the external conditions of the video. For example, the yellow marker tracking required no yellow coloured objects in the video frame (i.e.., background), unless the object is considerably smaller than the marker and can be masked or filtered out. In addition, the lighting, distance of the camera from the marker, the type of marker all influenced the detection ability which often resulted in a need for tuning on a trial by trial basis. In addition, the colour histogram thresholds may change depending on the type of yellow marker used which requires consistency in the marker and the lighting conditions.

Following the marker tracking, markerless alternatives were explored. The first markerless hand fist tracking explored was the bounding box method. While hand fist bounding box tracking did not meet the required accuracy constraints, it met several other requirements such as being rapid, robust, and easy to use. The three main reasons that likely contributed to the poor performance were false detections, motion blur, and inconsistent tracking point. Since the bounding box changes shape and height depending on the perspective of the hand shown, such that when the hand is at the top of the image frame, it is likely to be smaller than while at the bottom of the image frame. Conversely, when the top of the hand is more visible, the box is likely to be bigger. This affects the centroid and the tracking, causing inconsistencies in the point being tracked. The bounding box method is unlikely to yield the required accuracy constraints for this problem. Therefore, a more complex method was explored.

The fist tracking method using hand landmark detection and tracking produced improved performance compared to bounding box. This is likely attributable to finer resolution while tracking a single landmark on the hand throughout the video, regardless of the hand perspective. This method was the most promising thus far in terms of meeting the outlined constraints. While it did not satisfy all the accuracy constraints, it was the most appropriate method, and refinements to this method were implemented to improve accuracy performance. One issue that contributed to poor performance is that a hand fist is likely not a common hand pose in the hand image dataset used for training. As such, using the same hand landmark method, finger tracking was performed instead.

Tracking fingers from the front view using landmark detection was not possible as finger segmentation in that perspective was unreliable. Tracking the finger from a sagittal view prevents the ability to detect the second hand due to occlusions. As such, the camera was placed at ~45 degree angle relative to the sagittal plane to accommodate both these issues. Tracking the index finger instead of the fist significantly improved performance. However, the error in peak velocities was still higher than the allowable error with the lowest error being 232.97 (308.77) mm/s for the mean (SD), largely attributed to motion blur which introduced noise to the data.

Further modifications that were incorporated were increasing the sampling rate to 120 and 240 fps, which yielded significantly improved performance and met the constraints set for this problem. Remaining issues pertaining to the higher framerate results were a consequence of calibration problems. The only 120 fps trial which yielded an error in peak velocity value higher than the allowable limit was due to issues in the calibration procedure. The error is attributed to the participant shifting their position after the calibration video resulting in a different distance from the camera for the trial video.

In conclusion, based on the obtained results, the 120 fps video is the optimal option for this problem. Certain measures need to be taken to ensure calibration and trial settings are performed optimally to avoid performance issues, which are highlighted in Table 20 in Appendix

B. Overall, the performance of the 120 fps video meets the requirements for this problem in terms of its position estimates. Features other than the peak velocity are recommended for the final model as the computer vision methods seems to be lacking in performance for velocity estimates. It is recommended that the final features be selected from computer vision data collected for the proposed paradigm, from concussion and healthy participants (rather than the features currently selected from motion capture data). If velocity estimates are needed, the 240 fps video would yield better performance.

In terms of the calibration procedure, the main limitation is the angle introduced by vertical arm movement. Due to the nature of the arm joint, the vertical movements of the arm will not produce a pure vertical motion, as such, at the top and bottom of the motion, the real pure vertical distance of the squares will be slightly different than the constant value used. This results in a slight angle of the calibration board at the top and bottom of the calibration range, leading to an inaccurate vertical real world distance and overestimation of the distances. The pixel distance at the top and bottom of the range are expected to vary due to the perspective of the camera, i.e., the camera is oriented in a way such that it is slightly angled downwards. As such, these changes must be accounted for. For the first calibration procedure, the pixel distances at the top and bottom of the range were averaged to produce a constant calibration ratio. The averaging likely reduced the effect of the real world distance inaccuracy problem. For the calibration curve, the real to pixel distance ratio at the top and bottom of the range were used to produce a curve, such that, depending on the location of the hands in the image frame, there were multiplied by a different calibration value. The results from using this method did not show consistent improvement over the calibration constant, likely due to the error in real world distance described earlier, which is amplified in the case of the calibration curve. A potential next step would be to attempt to force the board to be straight at the top and bottom of the range. However, this will make calibration more difficult and will require more input from the collector and the user. The main limitation of the computer vision method is calibration, further work into improving the calibration procedure to make it more robust, and improve ease of calibrating each hand separately would be the focus of the next steps. In conclusion, the calibration constant is currently the optimal option.

4 Conclusion and Limitations

After experiencing a concussion, individuals continuing to show symptoms display altered performance in a bimanual coordination task compared to peers without history of concussion. Currently, concussion screening tools employed on the sidelines are subjective in nature and rely on human judgment. This presents a major drawback especially when these subjective methods are used to assess motor function, such as bimanual coordination. To attain more sensitive measures of motor impairment due to concussion, this work proposes a novel bimanual motor control testing paradigm, which is feasible to implement on the sidelines. This was shown to be effective in identifying concussion, as per our proof-of-concept analyses performed in Chapter 2.

The findings from Chapter 2 demonstrate a moderate-to-strong ability to detect individuals with a concussion (i.e., CON-S vs HC) using kinematic metrics of bimanual coordination (Accuracy = 86-88%; AUROC = 0.82-0.87). However, asymptomatic individuals (CON-A) did not exhibit the same motor deficits as symptomatic individuals (with a recent diagnosis). While the study yielded promising results, next steps would be to collect more data, and implement the same analyses shown in Chapter 2 on the larger dataset. This is necessary to ensure that the results shown in this work can be extended to a larger sample size, and that upon obtaining more samples (more participants), the same patterns are still attained.

Following larger sample studies, a potentially interesting avenue to explore is individualizing this assessment. First, a study must be conducted to assess the need for individualization, by comparing performance of different athletes on the selected metrics at baseline, and then comparing the outcome to the normative-data based classifier built in this work. One drawback of individualized assessment is the need for habitual, serial data collection, to construct individualized profiles, which reduces the feasibility of this tool as a rapid assessment method. One potential advantage is increased sensitivity to motor impairment. On the other hand, using normative data to build the classifier, but incorporating information about sex, age and BMI as features might help personalize predictions without the need for individualized data. This will need to be incorporated into the data collection procedure and analyzed to see if these groups show statistically significant differences in performance.

Motion capture was used in Chapter 2 for collecting the position data employed for analyzing the proposed objective assessment, diminishing the feasibility of this method as a rapid screening tool to be used on the sidelines. To implement this procedure into sideline assessment, a rapid, portable, low-cost, and easy-to-use data collection system is required. For the second contribution of this thesis, a tool to tackle this issue is proposed in Chapter 3 by employing computer vision techniques for collecting the required position data using a cellphone camera.

The proposed computer vision system is designed to be an easy-to-use, mobile sideline concussion screening tool, which would reduce cost and complexity of currently available data collection procedures . In Chapter 3, a proof of concept study was conducted to analyze the feasibility of using such a method for the proposed task, and showed promising results. Overall, the performance of the markerless finger-tracking at 45 degree angled view and 120 fps video meets the design requirements to estimate position. However, the evaluated computer vision methods are lacking in performance for velocity estimates. While the proposed markerless hand tracking method is fairly easy-to-use, proper collection settings are still required for good detection and tracking. The limitations imposed by the calibration procedure outlined in chapter 3 should be addressed to further improve performance. Calibration will likely be improved by obtaining 3D position information, rather than treating the motion as 2D. However, 3D data collection would require the use of more than one camera or a depth camera, which would reduce the rapidity and ease-of-use; therefore, a compromise must be made between accuracy and usability. Further work on improving the calibration procedure using the 2D representation would likely yield improvements in performance. This could include limiting the calibration task (motion) performed by the user or utilizing a different reference object. Speed could also be optimized for more rapid analysis. No attempt to improve speed was performed in this work. It should be noted that for some trials, namely the 60 fps trials, the checkerboard was not detected which required manually checking the pixel height of the middle square from a video frame. Improving checkerboard detection presents another area of improvement, for better calibration.

Further areas to explore include investigating other bimanual coordination motions (for ex., of the lower limbs) which could be used in conjunction with the paradigm proposed in this work to improve impairment detection. An alternative could be to incorporate the predictions of the proposed paradigm with the currently accepted FTN test results (performed by a human assessor). Such a hybrid approach could allow for higher acceptance rate of the proposed method in clinical settings. In addition, following the collection of more data, more complex models could be explored, such as deep learning approaches. The utility of fuzzy logic should also be explored to account for the imprecision and uncertainty that is often accompanied with medical diagnoses. Furthermore, it would be interesting to explore the use of multiple cameras (dual camera smartphones) to potentially improve video calibration and finger tracking.

Overall, this thesis presents work to identify discriminating features to facilitate objective motor coordination screening and a vision-based method to estimate these metrics. As proof-of-concept work, further development is needed. Collecting a larger dataset from participants with and without a concussion using the finger landmark tracking method, and applying this data to determine a new set of optimal features will likely yield improved performance. Identifying new features might allow for the simpler alternatives (e.g., 60 fps video) to be used, wherein features that the computer vision system was unable to recreate with high quality may be substituted by other features which are better distinguished using the landmark tracking method. In other words, implementing the same experimental procedure, feature extraction methods, and model building methods described in Chapter 2, but with data collected from the computer vision system rather than motion capture (e.g., Optotrak), would offer a training dataset towards identifying the most descriptive feature set to suit the computer vision data specifically. This approach would inform the continued development of a motor assessment tool to screen for concussion with stronger sensitivity and reliability. This thesis contributes towards the development of new assessment methods to inform diagnoses, and develop rehabilitation strategies to restore the body and mind to pre-concussion capabilities, if possible.

References

[1] R. Saffary, L. Chin and R. Cantu, "Sports Medicine", American Journal of Lifestyle Medicine, vol. 6, no. 2, pp. 133-140, 2011. Available: 10.1177/1559827611411649.

[2] What is ImPACT? Information For Test Takers, Parents and Guardians. Impact Applications, Inc.

[3] H. Belanger, J. Uomoto and R. Vanderploeg, "The Veterans Health Administration System of Care for Mild Traumatic Brain Injury", Journal of Head Trauma Rehabilitation, vol. 24, no. 1, pp. 4-13, 2009. Available: 10.1097/htr.0b013e3181957032.

[4] N. Elsayed, "Toxicology of blast overpressure", Toxicology, vol. 121, no. 1, pp. 1-15, 1997. Available: 10.1016/s0300-483x(97)03651-2.

[5] A. Ommaya and A. Hirsch, "Tolerances for cerebral concussion from head impact and whiplash in primates", Journal of Biomechanics, vol. 4, no. 1, pp. 13-21, 1971. Available: 10.1016/0021-9290(71)90011-x.

[6] C. McKeever and P. Schatz, "Current Issues in the Identification, Assessment, and Management of Concussions in Sports-Related Injuries", Applied Neuropsychology, vol. 10, no. 1, pp. 4-11, 2003. Available: 10.1207/s15324826an1001_2.

[7] C. Giza and J. Kutcher, "An Introduction to Sports Concussions", CONTINUUM: Lifelong Learning in Neurology, vol. 20, pp. 1545-1551, 2014. Available: 10.1212/01.con.0000458975.78766.11.

[8] B. Asken, M. McCrea, J. Clugston, A. Snyder, Z. Houck and R. Bauer, "“Playing Through It”: Delayed Reporting and Removal From Athletic Activity After Concussion Predicts Prolonged Recovery", Journal of Athletic Training, vol. 51, no. 4, pp. 329-335, 2016. Available: 10.4085/1062-6050-51.5.02.

[9] B. Asken et al., "Immediate Removal From Activity After Sport-Related Concussion Is Associated With Shorter Clinical Recovery and Less Severe Symptoms in Collegiate Student-Athletes", The American Journal of Sports Medicine, vol. 46, no. 6, pp. 1465-1474, 2018. Available: 10.1177/0363546518757984.

[10] P. Schatz, J. Pardini, M. Lovell, M. Collins and K. Podell, "Sensitivity and specificity of the ImPACT Test Battery for concussion in athletes", Archives of Clinical Neuropsychology, vol. 21, no. 1, pp. 91-99, 2006. Available: 10.1016/j.acn.2005.08.001.

[11] B. Asken et al., "Immediate Removal From Activity After Sport-Related Concussion Is Associated With Shorter Clinical Recovery and Less Severe Symptoms in Collegiate Student-Athletes", The American Journal of Sports Medicine, vol. 46, no. 6, pp. 1465-1474, 2018. Available: 10.1177/0363546518757984.

[12] M. Barnhart, R. Bay and T. Valovich McLeod, "The Influence of Timing of Reporting and Clinic Presentation on Concussion Recovery Outcomes: A Systematic Review and Meta-Analysis", Sports Medicine, vol. 51, no. 7, pp. 1491-1508, 2021. Available: 10.1007/s40279-021-01444-7.

[13] P. McCrory et al., "Consensus Statement on Concussion in Sport: the 3rd International Conference on Concussion in Sport held in Zurich, November 2008", 2021.

[14] A. Maerlender et al., "Examination of the Construct Validity of Impact™ Computerized Test, Traditional, and Experimental Neuropsychological Measures", The Clinical Neuropsychologist, vol. 24, no. 8, pp. 1309-1325, 2010. Available: 10.1080/13854046.2010.516072.

[15] R. Echemendia et al., "The Sport Concussion Assessment Tool 5th Edition (SCAT5)", British Journal of Sports Medicine, pp. bjsports-2017-097506, 2017. Available: 10.1136/bjsports-2017-097506.

[16] "Exercise Database – SCAT3 Finger-to-nose (FTN) Assessment", Fitnessandhealthpromotion.ca. [Online]. Available: https://fitnessandhealthpromotion.ca/exercise-database/entry/55594/.

[17] M. Scott Moses, "Finger-to-Nose Test", Fpnotebook.com, 2021. [Online]. Available: https://fpnotebook.com/er/exam/FngrTNsTst.htm.

[18] B. Kashyap, D. Phan, P. Pathirana, M. Horne, L. Power and D. Szmulewicz, "Objective Assessment of Cerebellar Ataxia: A Comprehensive and Refined Approach", Scientific Reports, vol. 10, no. 1, 2020. Available: 10.1038/s41598-020-65303-7.

[19] C. Mang, T. Whitten, M. Cosh, S. Dukelow and B. Benson, "Assessment of Postural Stability During an Upper Extremity Rapid, Bimanual Motor Task After Sport-Related Concussion", Journal of Athletic Training, vol. 55, no. 11, pp. 1160-1173, 2020. Available: 10.4085/1062-6050-378-19.

[20] E. Stueland, "The Effects of Mild Concussion on Rapid Alternating Movement, as Indexed by Measure of Reaction Time and Motor Coordination", Ph.D, California School of Professional Psychology - San Diego, 2001.

[21] S. Huang, "The Roles of Visual and Kinesthetic Information in Learning and Control of Bimanual Coordination", Ph.D, University of Wyoming, 2020.

[22] J. Yue, R. Phelps, A. Chandra, E. Winkler, G. Manley and M. Berger, "Sideline Concussion Assessment: The Current State of the Art", Neurosurgery, vol. 87, no. 3, pp. 466-475, 2020. Available: 10.1093/neuros/nyaa022.

[23] L. King et al., "Sensor-Based Balance Measures Outperform Modified Balance Error Scoring System in Identifying Acute Concussion", Annals of Biomedical Engineering, vol. 45, no. 9, pp. 2135-2145, 2017. Available: 10.1007/s10439-017-1856-y.

[24] S. Broglio, K. Guskiewicz and J. Norwig, "If You're Not Measuring, You're Guessing: The Advent of Objective Concussion Assessments", Journal of Athletic Training, vol. 52, no. 3, pp. 160-166, 2017. Available: 10.4085/1062-6050-51.9.05.

[25] M. Zhu, Z. Huang, C. Ma and Y. Li, "An Objective Balance Error Scoring System for Sideline Concussion Evaluation Using Duplex Kinect Sensors", Sensors, vol. 17, no. 10, p. 2398, 2017. Available: 10.3390/s17102398.

[26] R. Ventura, L. Balcer, S. Galetta and J. Rucker, "Ocular motor assessment in concussion: Current status and future directions", Journal of the Neurological Sciences, vol. 361, pp. 79-86, 2016. Available: 10.1016/j.jns.2015.12.010.

[27] A. R. Scherger, D. Gonzalez, J. Y. Tung, R. J. Ibley, E. A. Roy, “The Influence of a Weighted Perturbation on a Bimanual Coordination Task,” Department of Kinesiology and Health Sciences, 2014.

[28] A. Tapper, D. Gonzalez, E. Roy and E. Niechwiej-Szwedo, "Executive function deficits in team sport athletes with a history of concussion revealed by a visual-auditory dual task paradigm", Journal of Sports Sciences, vol. 35, no. 3, pp. 231-240, 2016. Available: 10.1080/02640414.2016.1161214.

[29] R. Moore, S. Broglio and C. Hillman, "Sport-Related Concussion and Sensory Function in Young Adults", Journal of Athletic Training, vol. 49, no. 1, pp. 36-41, 2014. Available: 10.4085/1062-6050-49.1.02.

[30] F. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform", Proceedings of the IEEE, vol. 66, no. 1, pp. 51-83, 1978. Available: 10.1109/proc.1978.10837.

[31] J. Candy, "Multitaper Spectral Estimation: An Alternative to the Welch Periodogram Approach", 2019. Available: 10.2172/1560107.

[32] A. Chave, "A multitaper spectral estimator for time-series with missing data", Geophysical Journal International, vol. 218, no. 3, pp. 2165-2178, 2019. Available: 10.1093/gji/ggz280.

[33] D. Thomson, "Spectrum estimation and harmonic analysis", Proceedings of the IEEE, vol. 70, no. 9, pp. 1055-1096, 1982. Available: 10.1109/proc.1982.12433.

[34] A. Chave, "Mwps(xx,varargin)", MATLAB Central File Exchange, 2021. [Online]. Available:https://www.mathworks.com/matlabcentral/fileexchange/65796-mwps-xx-varargin.

[35] P. Casas, "Feature Selection using Genetic Algorithms in R", Medium, 2019. Available: https://towardsdatascience.com/feature-selection-using-genetic-algorithms-in-r-3d9252f1aa66.

[36] M. Calzolari, "sklearn-genetic (Version 0.4.1) [Computer software]", 2021. Available: https://doi.org/10.5281/zenodo.4661248

[37] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd ed. O'Reilly Media, 2019.

[38] S. Dimitriadis, D. Liparas and ADNI, "How random is the random forest? Random forest algorithm on the service of structural imaging biomarkers for Alzheimer's disease: from Alzheimer's disease neuroimaging initiative (ADNI) database", Neural Regeneration Research, vol. 13, no. 6, p. 962, 2018. Available: 10.4103/1673-5374.233433.

[39] V. Alto, "Understanding AdaBoost for Decision Tree", Medium, 2020. [Online]. Available: https://towardsdatascience.com/understanding-adaboost-for-decision-tree-ff8f07d2851.

[40] R. Guo, Z. Zhao, T. Wang, G. Liu, J. Zhao and D. Gao, "Degradation State Recognition of Piston Pump Based on ICEEMDAN and XGBoost", Applied Sciences, vol. 10, no. 18, p. 6593, 2020. Available: 10.3390/app10186593.

[41] D. Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation", 2010.

[42] J. Choudhary, "What is Model Validation.", Medium, 2020. Available: https://medium.com/analytics-vidhya/what-is-model-validation-257686d0253e.

[43] D. Powell, S. Stuart, D. Fearn, S. Bowen, H. Steel, T. Jones, and A. Godfrey, "Wearables as objective tools in sport-related concussion: A protocol for more informed player management," Physiotherapy, vol. 107, 2020.

[44] L. A. King, M. Mancini, P. C. Fino, J. Chesnutt, C. W. Swanson, S. Markwardt, and J. C. Chapman, "Sensor-based balance measures outperform modified balance error scoring system in identifying acute concussion," Annals of Biomedical Engineering, vol. 45, no. 9, pp. 2135–2145, 2017.

[45] D. A. Krause, S. E. Anderson, G. R. Campbell, S. J. Davis, S. W. Tindall, and J. H. Hollman, “Responsiveness of a balance assessment using a mobile application,” Sports Health: A Multidisciplinary Approach, vol. 12, no. 4, pp. 401–404, 2020.

[46] J. M. Kim, “A New Approach to Concussion Detection through Bimanual Coordination,” Department of Systems Design Engineering University of Waterloo, 2018.

[47] A. Nalci, A. Khodamoradi, O. Balkan, F. Nahab, and H. Garudadri, “A computer vision based candidate for functional balance test,” 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015.

[48] M. Zago, M. Luzzago, T. Marangoni, M. De Cecco, M. Tarabini, and M. Galli, “3D tracking of human motion using visual skeletonization and Stereoscopic Vision,” Frontiers in Bioengineering and Biotechnology, vol. 8, 2020.

[49] N. Nakano, T. Sakura, K. Ueda, L. Omura, A. Kimura, Y. Iino, S. Fukashiro, and S. Yoshioka, “Evaluation of 3D markerless motion capture accuracy using openpose with multiple video cameras,” Frontiers in Sports and Active Living, vol. 2, 2020.

[50] V. Dibia, “HandTrack: A Library For Prototyping Real-time Hand Tracking Interfaces using Convolutional Neural Networks”. https://github.com/victordibia/handtracking, 2021.

[51] “MediaPipe”. Google. https://github.com/google/mediapipe

[52] D. Giavarina, "Understanding Bland Altman analysis", Biochemia Medica, vol. 25, no. 2, pp. 141-151, 2015. Available: 10.11613/bm.2015.015.

[53] A. Kalra, "Decoding the Bland–Altman plot: Basic review", Journal of the Practice of Cardiovascular Sciences, vol. 3, no. 1, p. 36, 2017. Available: 10.4103/jpcs.jpcs_11_17.

[54] R. Klein, "Bland-Altman and Correlation Plot", MATLAB Central File Exchange, 2021. [Online]. Available:https://www.mathworks.com/matlabcentral/fileexchange/45049-bland-altman-and-correlation-plot.

[55] T. Chang Wei, G. Webb, F. Petitjean and P. Reichl, "Machine learning approaches for tamping effectiveness prediction", Conference: 2017 International Heavy Haul Association Conference (IHHA)At: Cape Town, South Africa, 2017.

[56] I. Oregi, A. Pérez, J. Del Ser and J. Lozano, "On-Line Dynamic Time Warping for Streaming Time Series", Machine Learning and Knowledge Discovery in Databases, pp. 591-605, 2017. Available: 10.1007/978-3-319-71246-8_36.

5 APPENDICES

5.1 Appendix A.

Computer Vision Hand Tracking Graphical Results

Colour tracking

<u>P1: Right Hand</u>

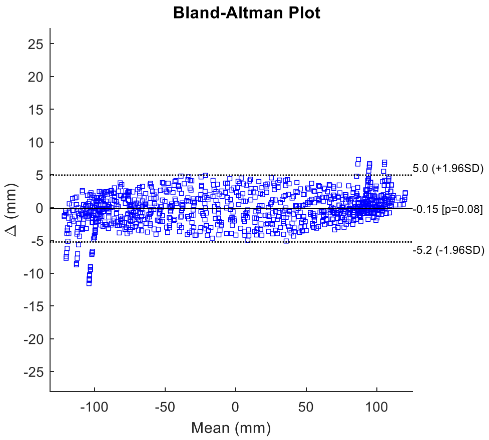


Figure 51: Bland Altman plot comparing position data obtained from colour tracking and Vicon motion capture

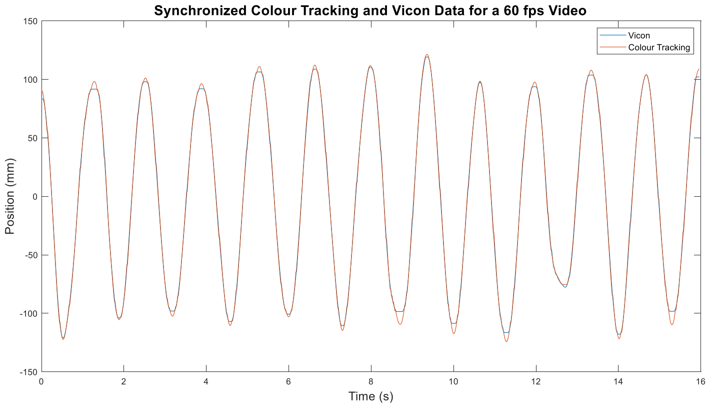


Figure 52: Synchronized colour tracking and Vicon position data using DTW

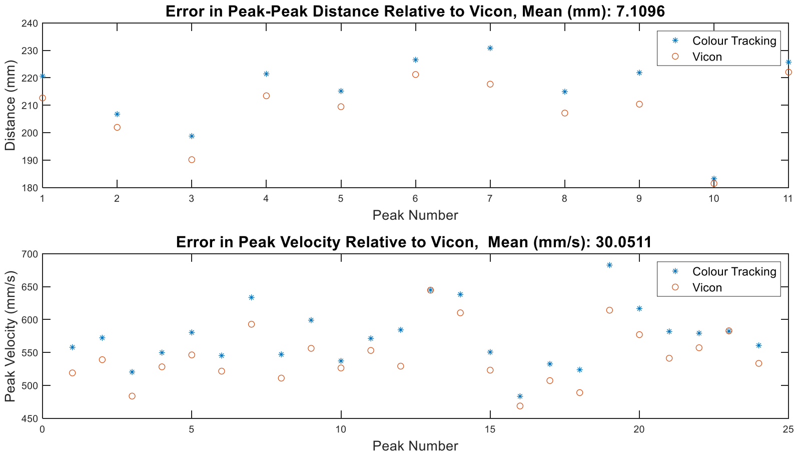


Figure 53: Top: Error in peak to peak distance between colour tracking and Vicon. Bottom: Error in peak velocity between colour tracking and Vicon

P2: Left Hand

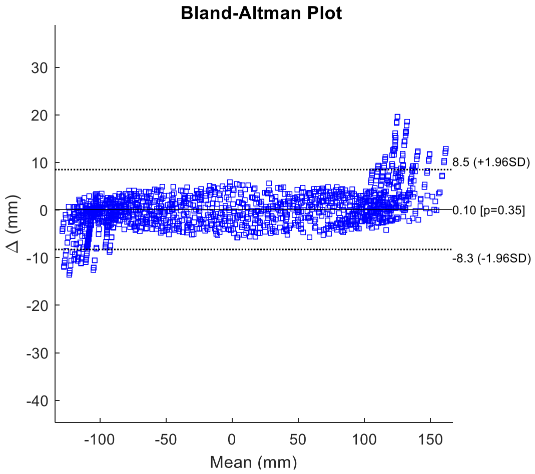


Figure 54: Bland Altman plot comparing position data obtained from colour tracking and Vicon motion capture

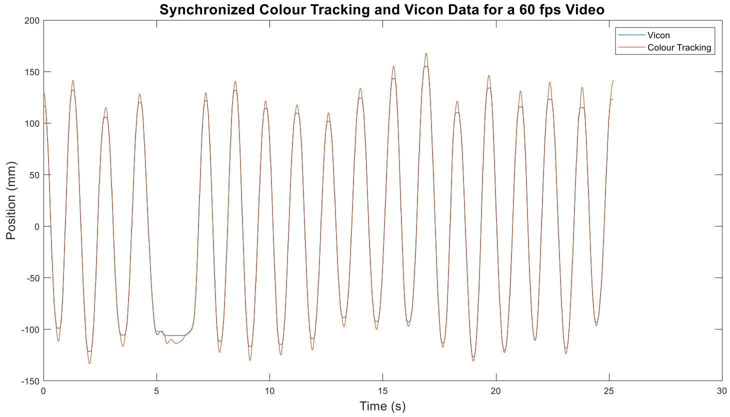


Figure 55: Synchronized colour tracking and Vicon position data using DTW

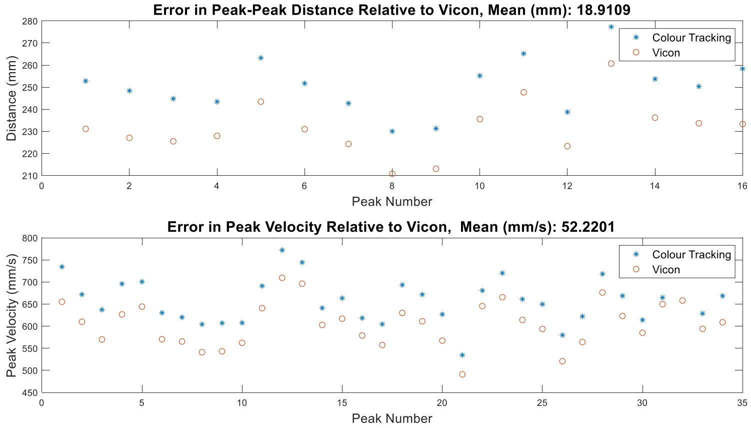


Figure 56: Top: Error in peak to peak distance between colour tracking and Vicon. Bottom: Error in peak velocity between colour tracking and Vicon

P2: Right Hand

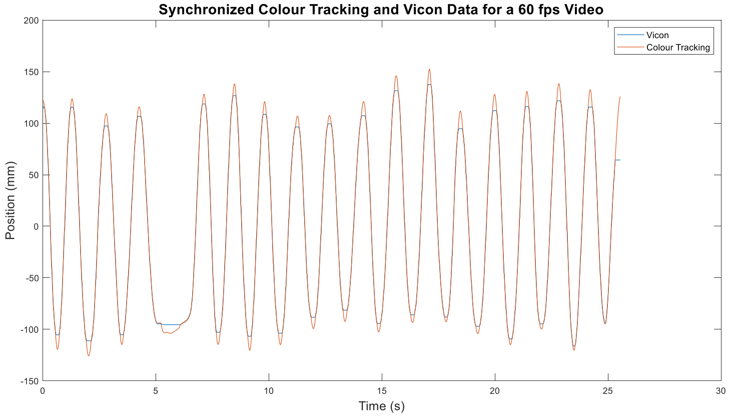


Figure 57: Synchronized colour tracking and Vicon position data using DTW

Bounding Box

P1: Right Hand

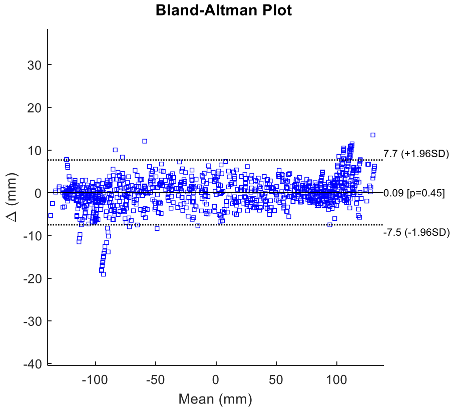


Figure 58: Bland Altman plot comparing position data obtained from Handtrack and Vicon motion capture

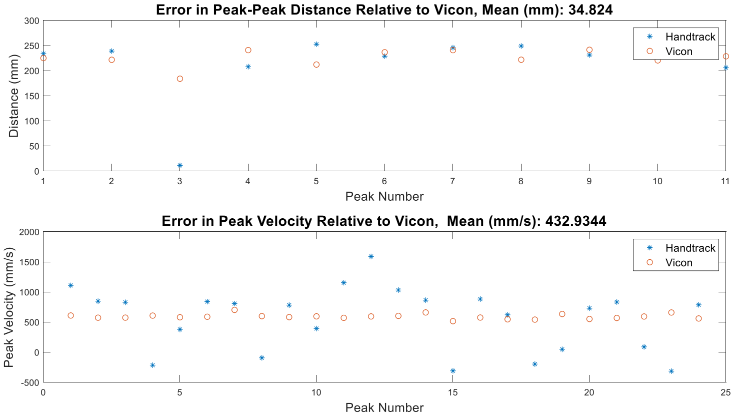


Figure 59: Top: Error in peak to peak distance between Handtrack and Vicon. Bottom: Error in peak velocity between Handtrack and Vicon

P1: Left Hand

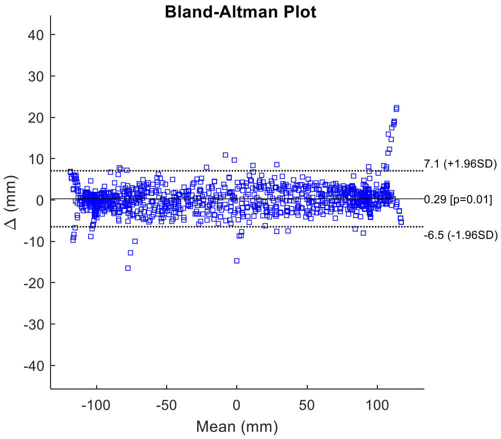


Figure 60: Bland Altman plot comparing position data obtained from Handtrack and Vicon motion capture

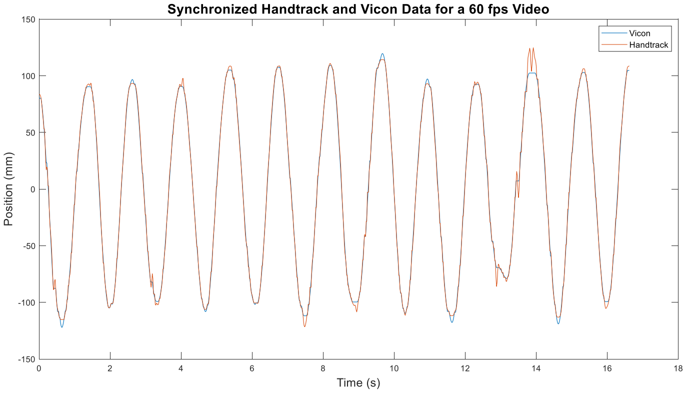


Figure 61: Synchronized Handtrack and Vicon position data using DTW

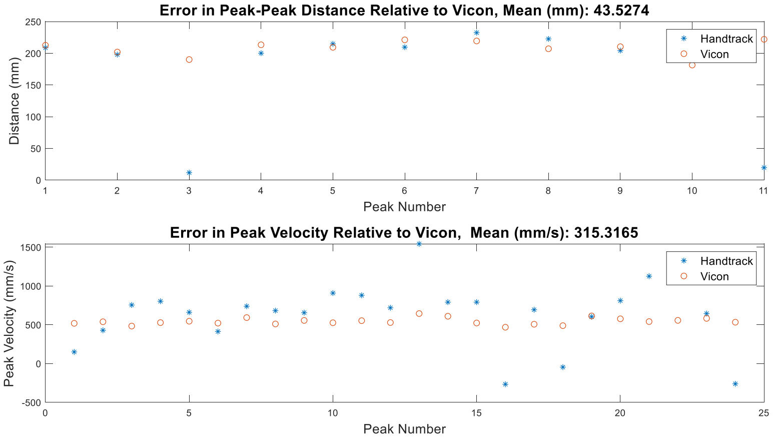


Figure 62: Top: Error in peak to peak distance between Handtrack and Vicon. Bottom: Error in peak velocity between Handtrack and Vicon

Landmark Tracking- Fist

P1: Right Hand

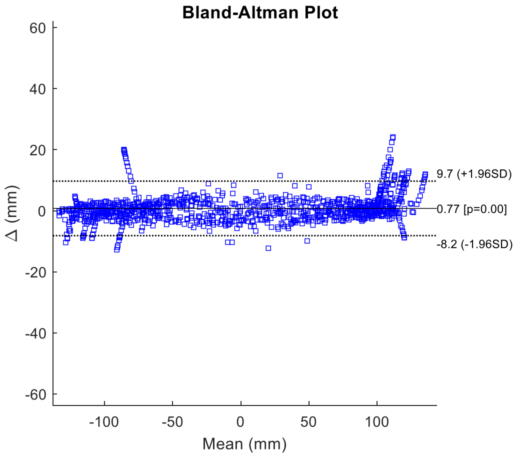


Figure 63: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture

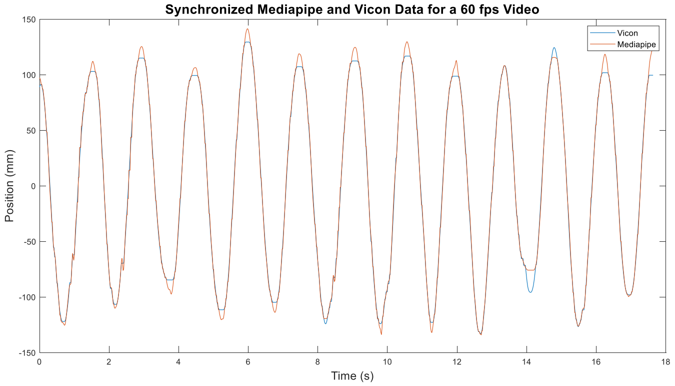


Figure 64: Synchronized Mediapipe and Vicon position data using DTW

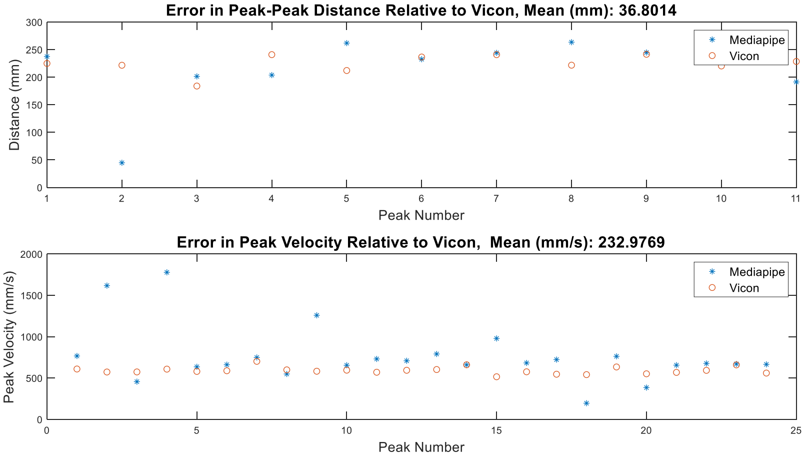


Figure 65: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

P2: Right Hand

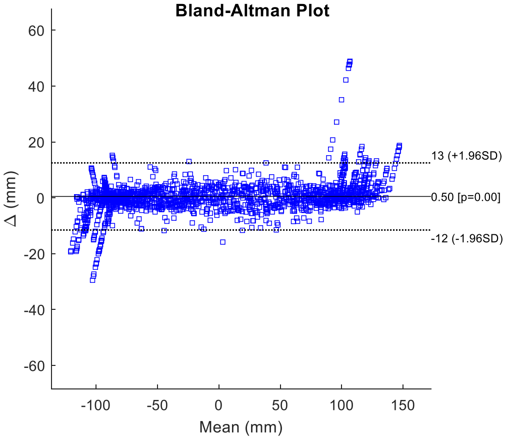


Figure 66: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture

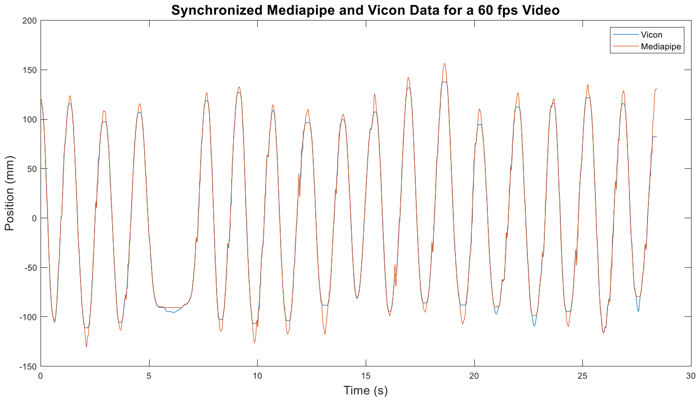
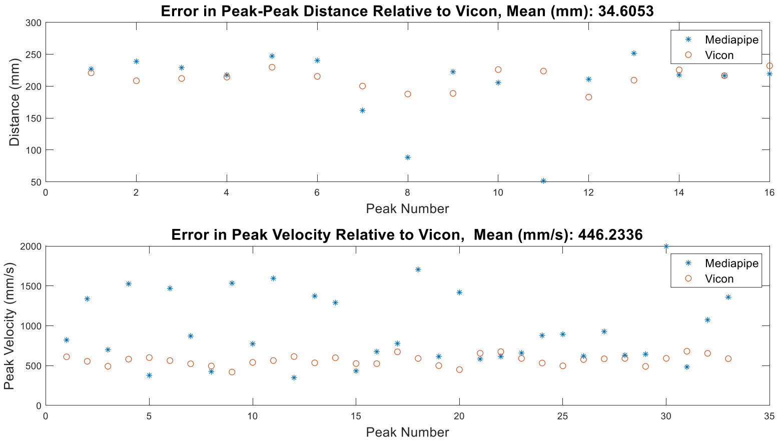


Figure 67: Synchronized Mediapipe and Vicon position data using DTW



Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

P2: Left Hand

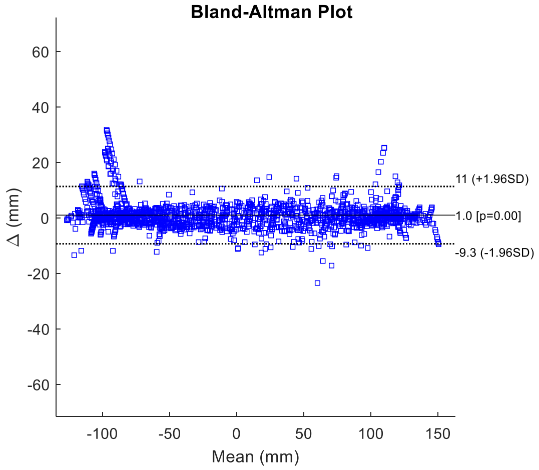


Figure 68: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture

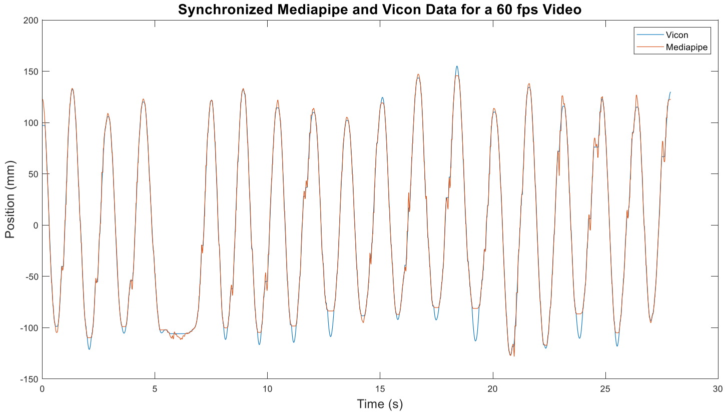


Figure 69: Synchronized Mediapipe and Vicon position data using DTW

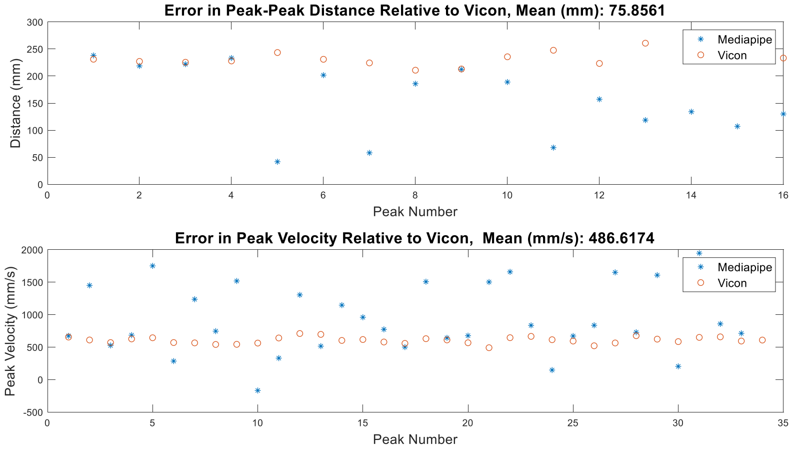


Figure 70: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

Finger Tracking- 60 fps

P1: Left Hand

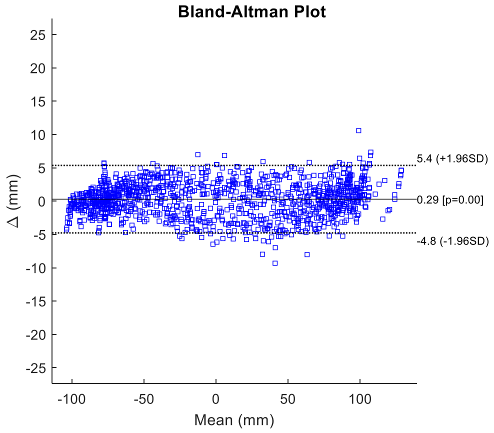


Figure 71: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture

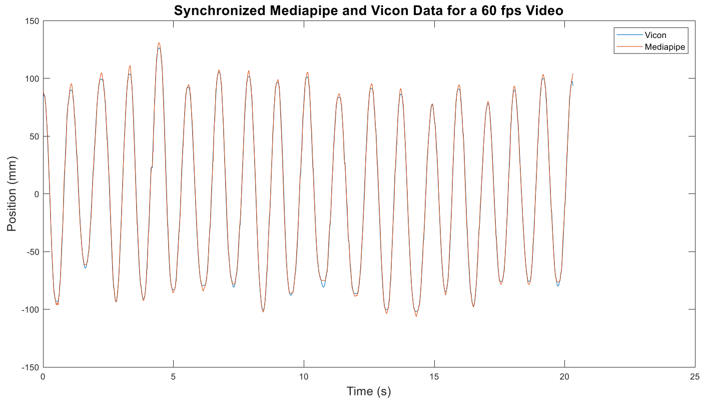


Figure 72: Synchronized Mediapipe and Vicon position data using DTW

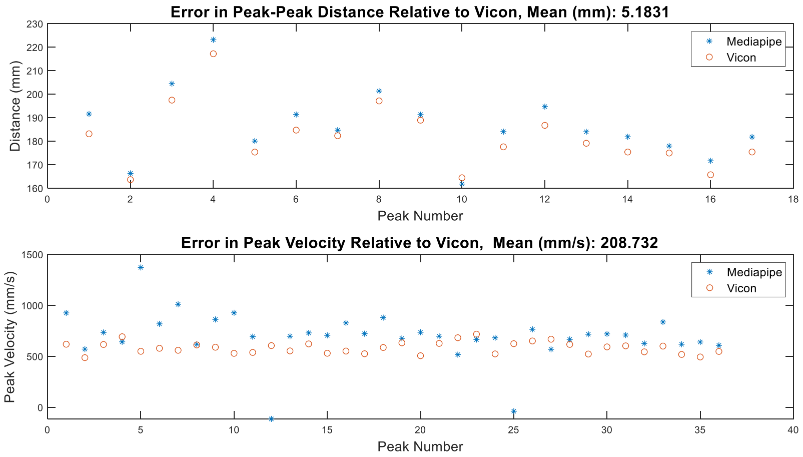


Figure 73: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

P2: Left Hand

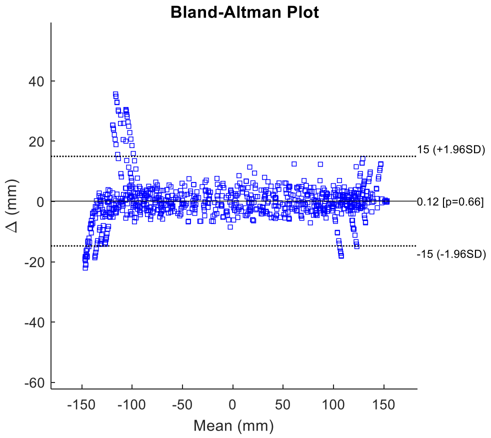


Figure 74: Bland Altman plot comparing position data obtain from Mediapipe and Vicon motion capture

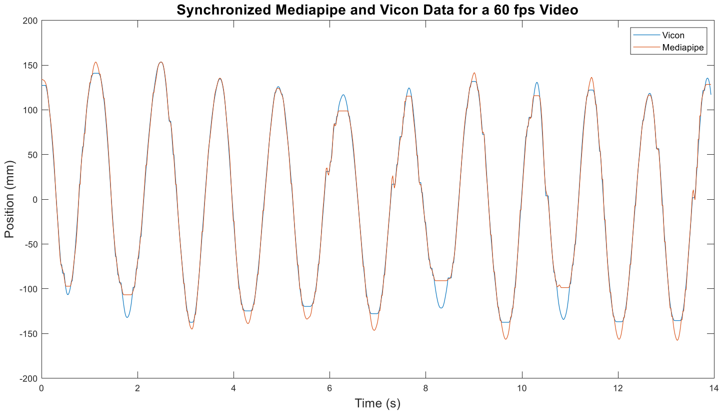


Figure 75: Synchronized Mediapipe and Vicon position data using DTW

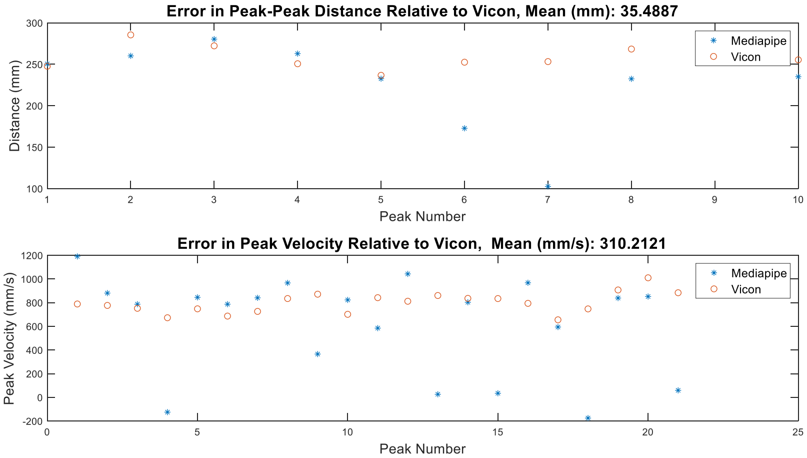


Figure 76: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

Finger Tracking- 120 fps

<u>P1: Left Hand</u>

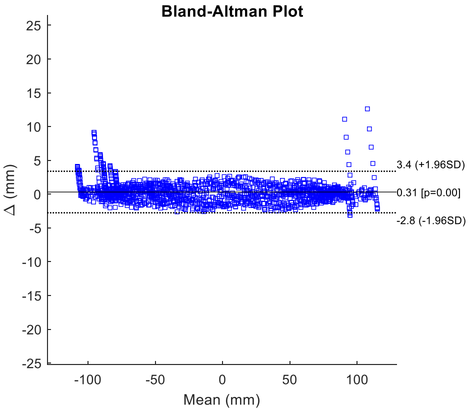


Figure 77: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 120 fps video

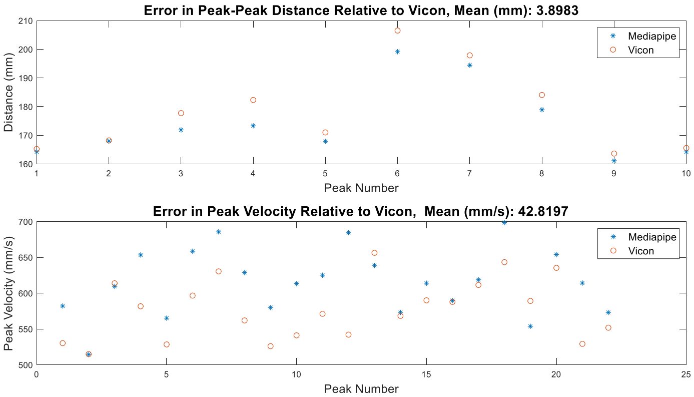


Figure 78: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between the two methods

P1: Right Hand

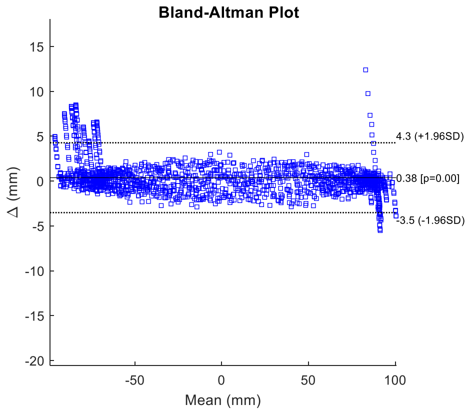


Figure 79: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 120 fps video

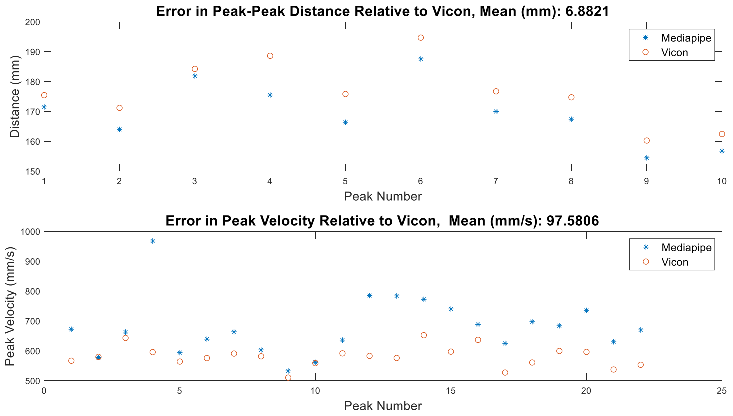


Figure 80: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

P2: Left Hand

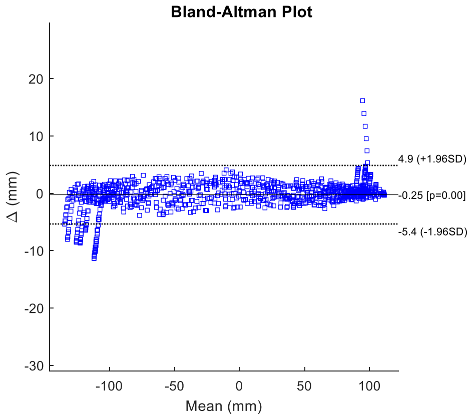


Figure 81: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 120 fps video

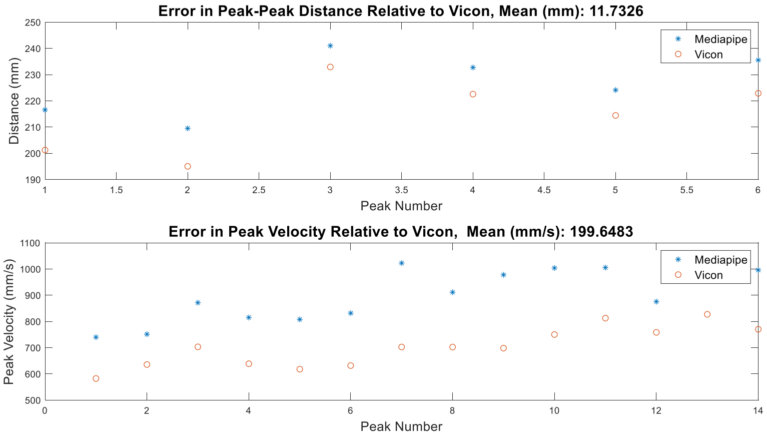


Figure 82: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

P2: Right Hand

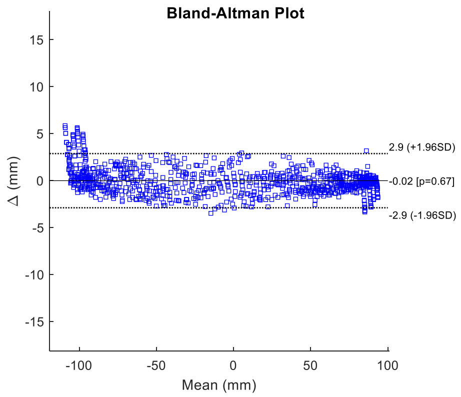


Figure 83: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 120 fps video

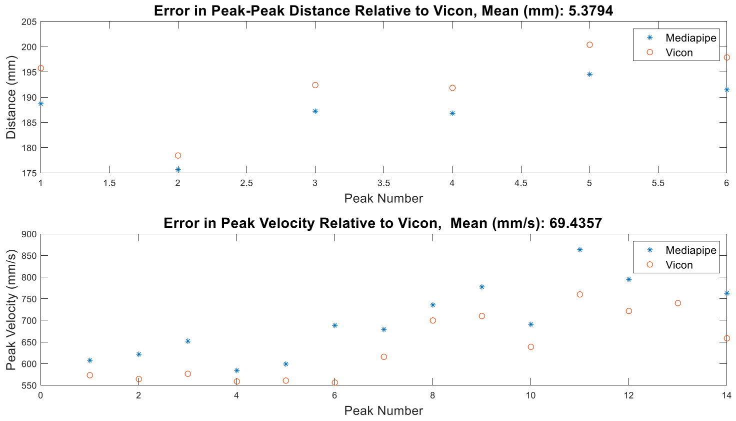


Figure 84: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

Landmark Tracking- 240 fps

<u>P1: Left Hand</u>

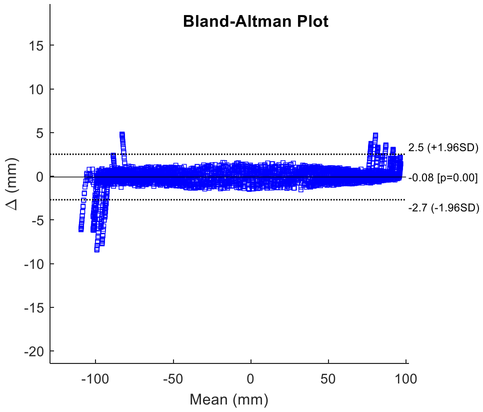


Figure 85: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 240 fps video

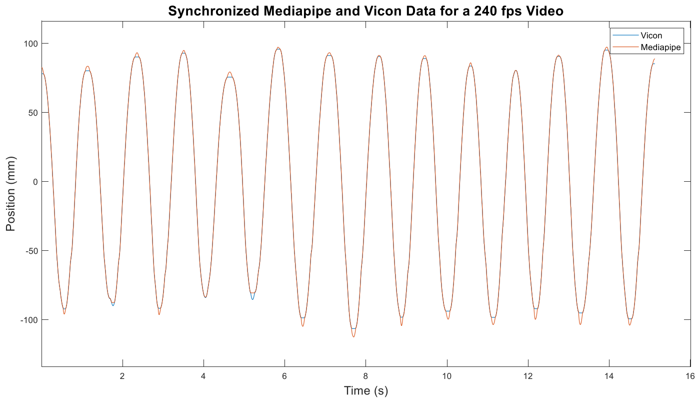


Figure 86: Synchronized Mediapipe and Vicon position data using DTW for a 240 fps video

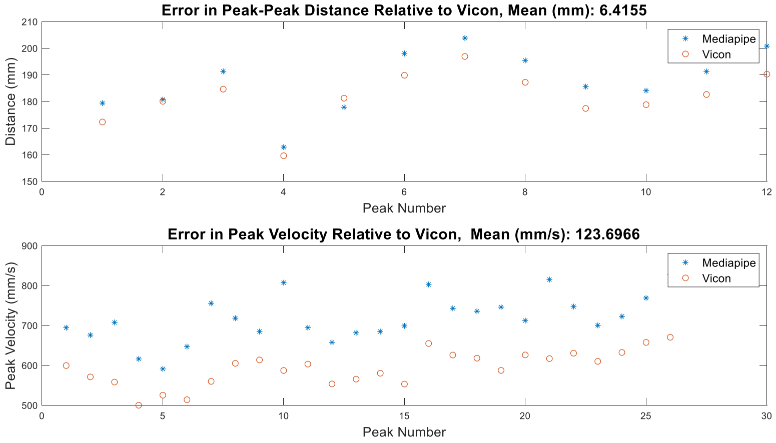


Figure 87: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

P2: Right hand

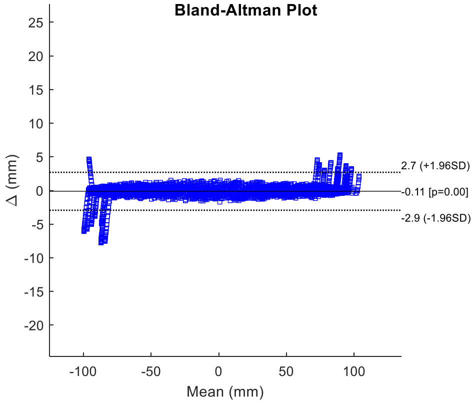


Figure 88: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 240 fps video

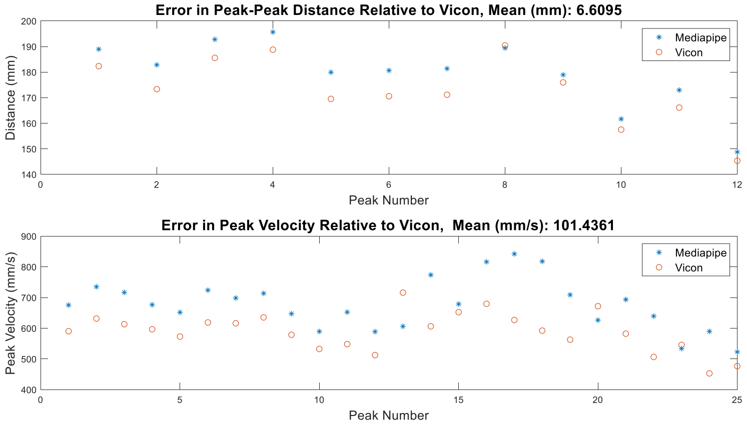


Figure 89: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

P2: Left Hand

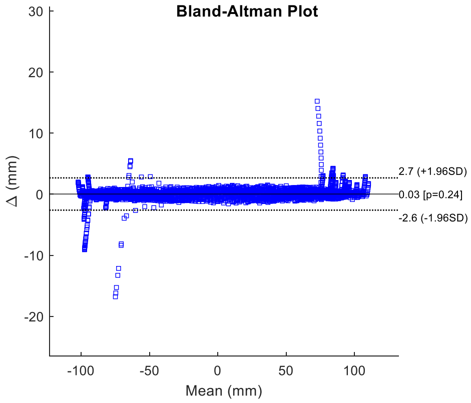


Figure 90: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 240 fps video

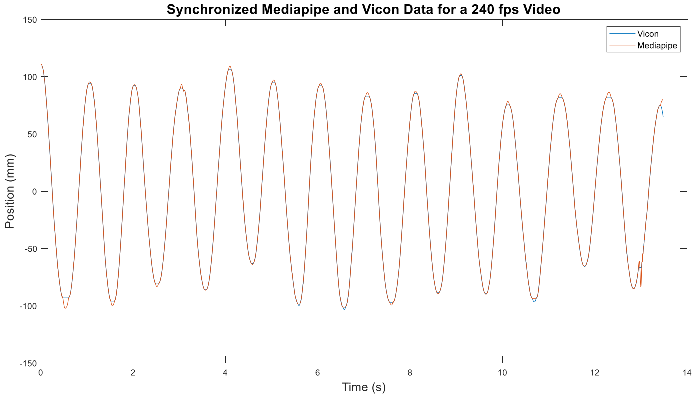


Figure 91: Synchronized Mediapipe and Vicon position data using DTW for a 240 fps video

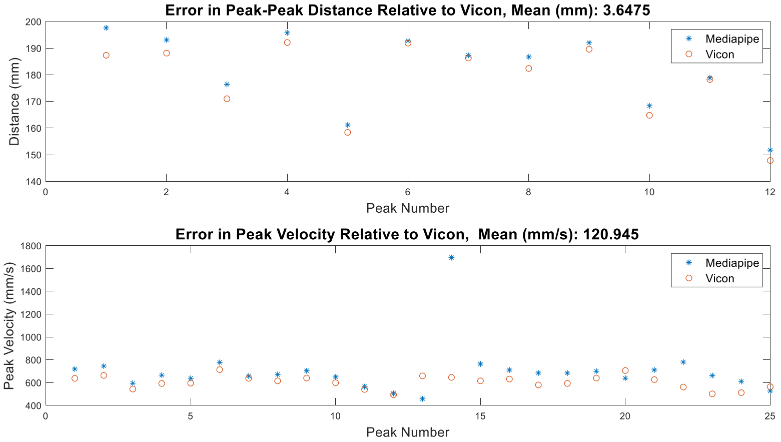


Figure 92: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

The graphical results for the calibration curve method are shown below. Results are shown for one participant, for each of the three tested frame rates.

P1-60 fps
<u>Right Hand</u>

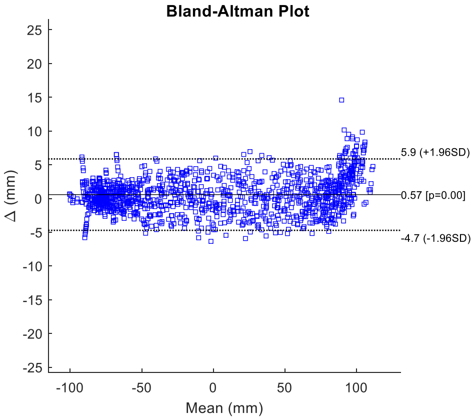


Figure 93: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 60 fps video

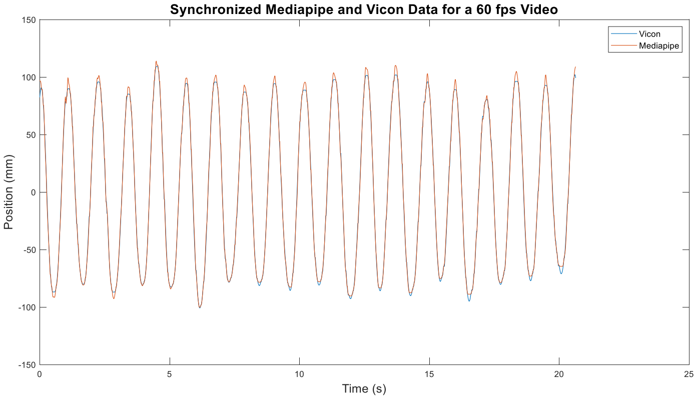


Figure 94: Synchronized Mediapipe and Vicon position data using DTW for a 60 fps video

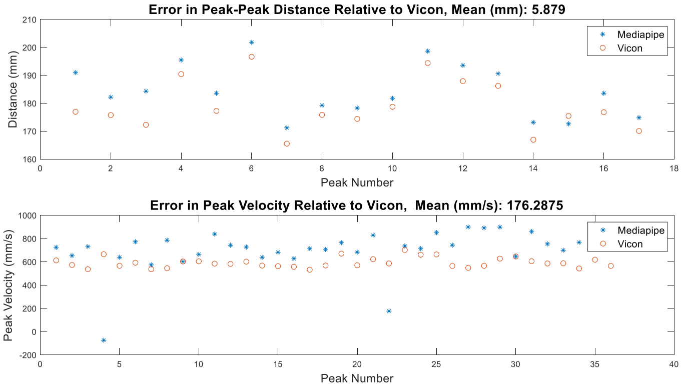


Figure 95: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

Left Hand

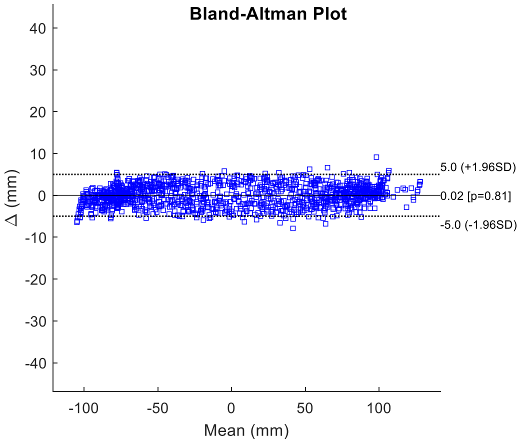


Figure 96: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 60 fps video

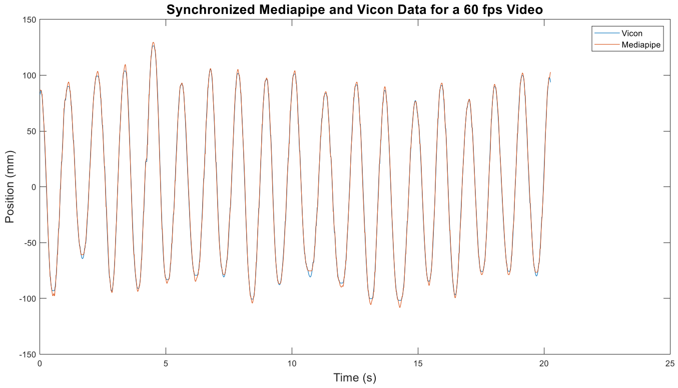


Figure 97: Synchronized Mediapipe and Vicon position data using DTW for a 60 fps video

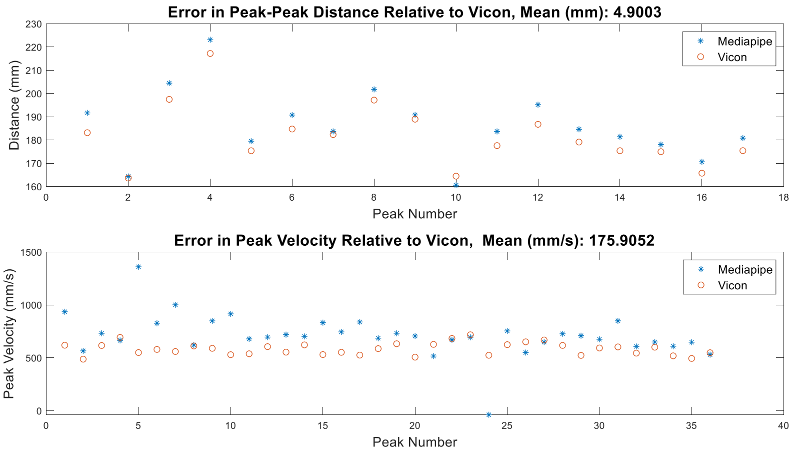


Figure 98: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

P2: Left Hand

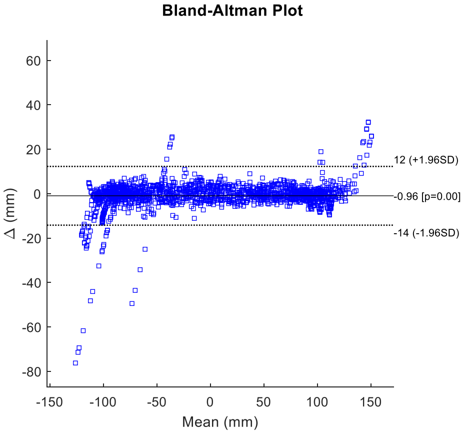


Figure 99: Bland Altman plot comparing position data obtained from Handtrack and Vicon motion capture

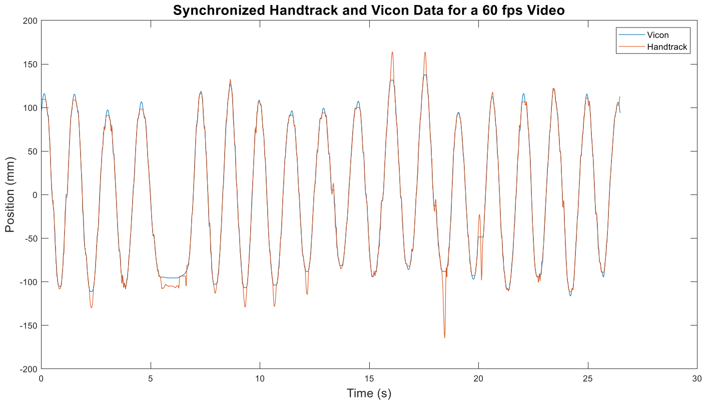


Figure 100: Synchronized Handtrack and Vicon position data using DTW

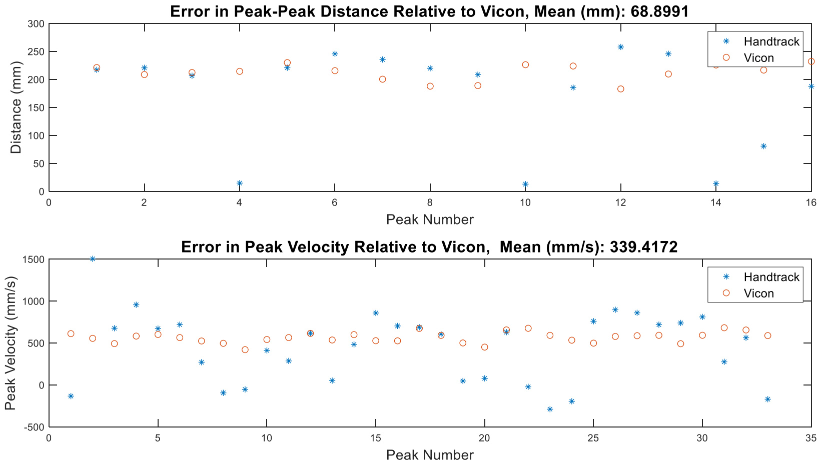


Figure 101: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

P1- 120 fps

<u>Right Hand</u>

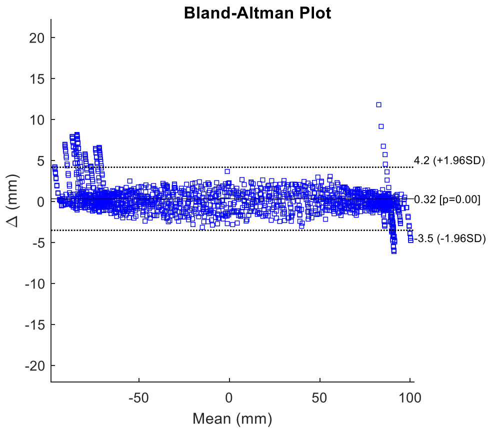


Figure 102: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 120 fps video

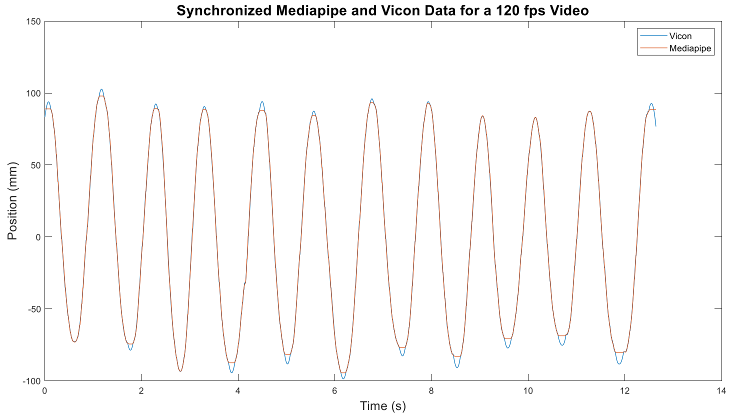


Figure 103: Synchronized Mediapipe and Vicon position data using DTW for a 120 fps video

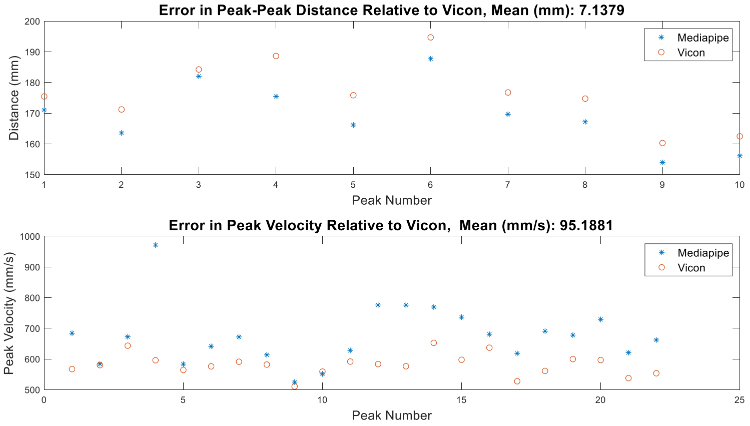


Figure 104: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

Left Hand

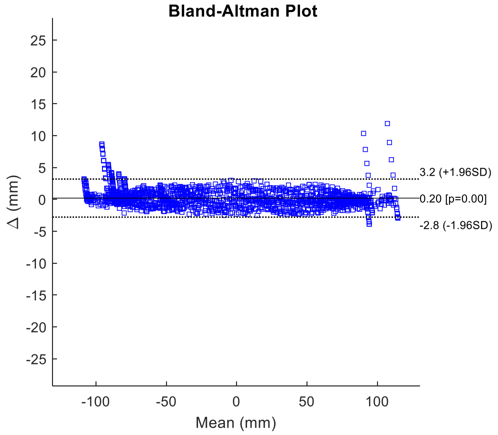


Figure 105: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 120 fps video

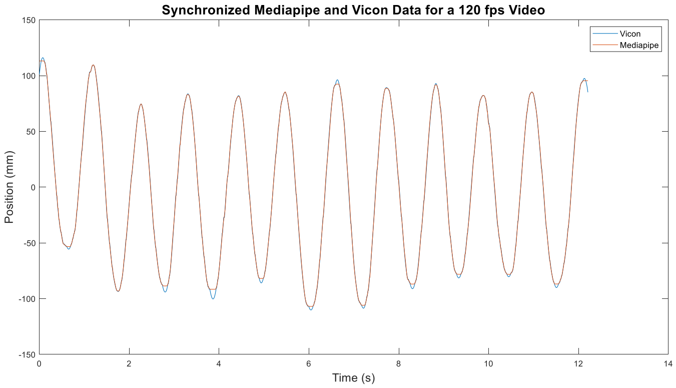


Figure 106: Synchronized Mediapipe and Vicon position data using DTW for a 120 fps video

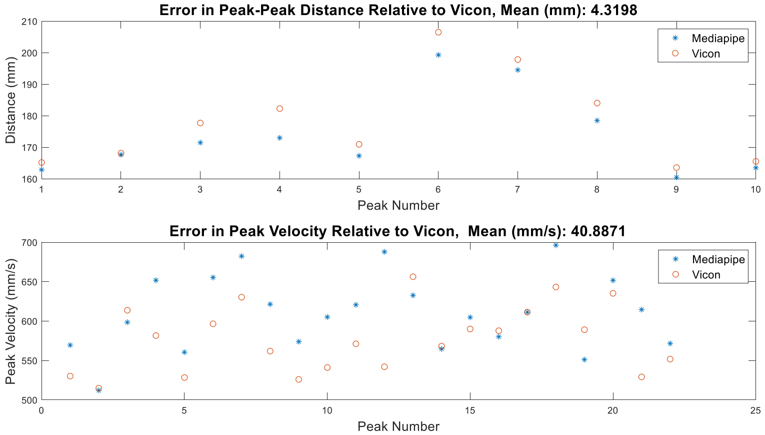


Figure 107: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

P2-240

Right Hand

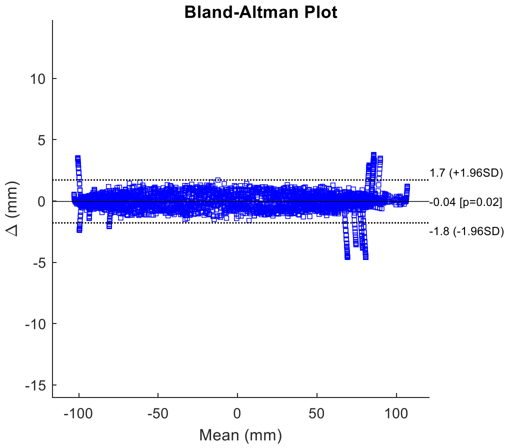


Figure 108: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 240 fps video

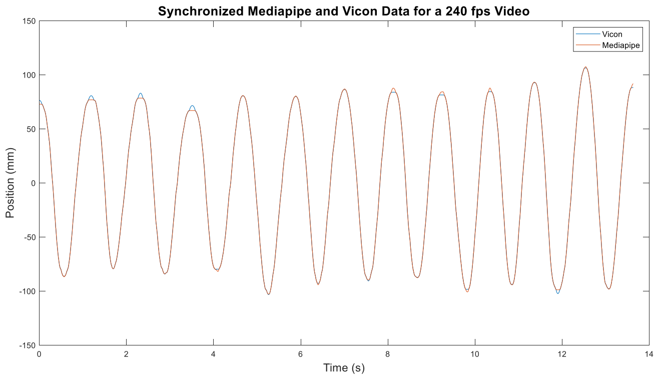


Figure 109: Synchronized Mediapipe and Vicon position data using DTW for a 240 fps video

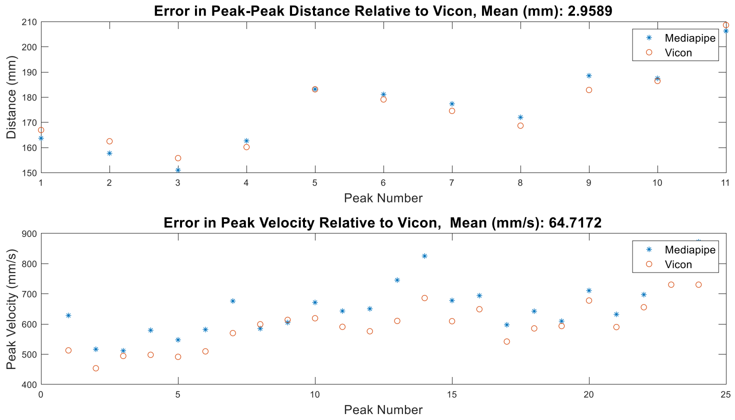


Figure 110: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

<u>Left Hand</u>

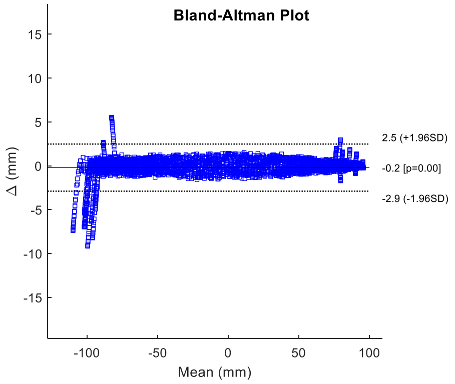


Figure 111: Bland Altman plot comparing position data obtained from Mediapipe and Vicon motion capture for a 240 fps video

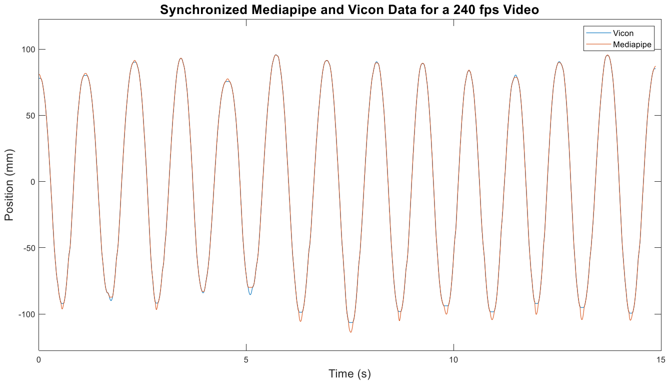


Figure 112: Synchronized Mediapipe and Vicon position data using DTW for a 240 fps video

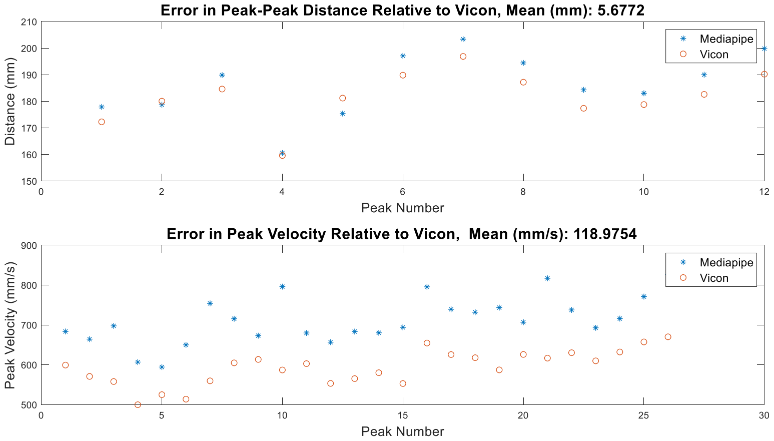
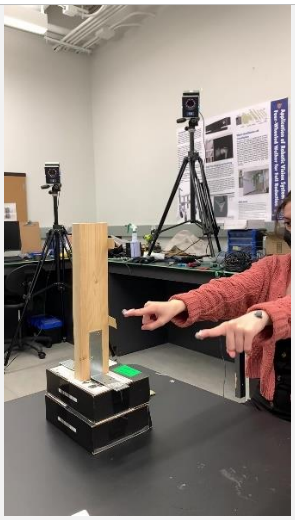
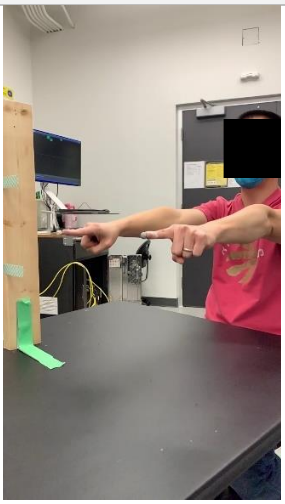


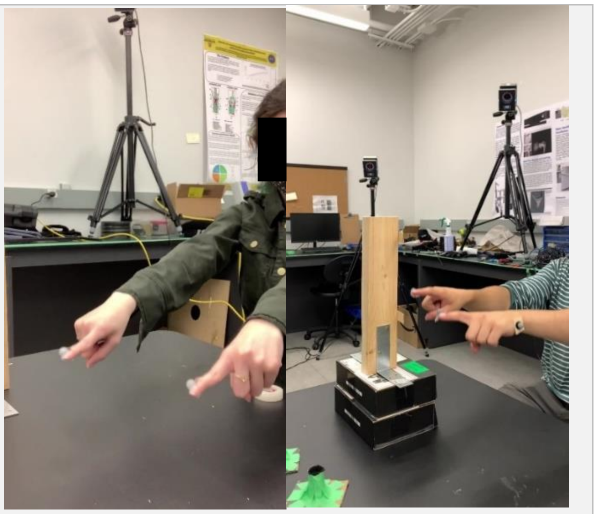
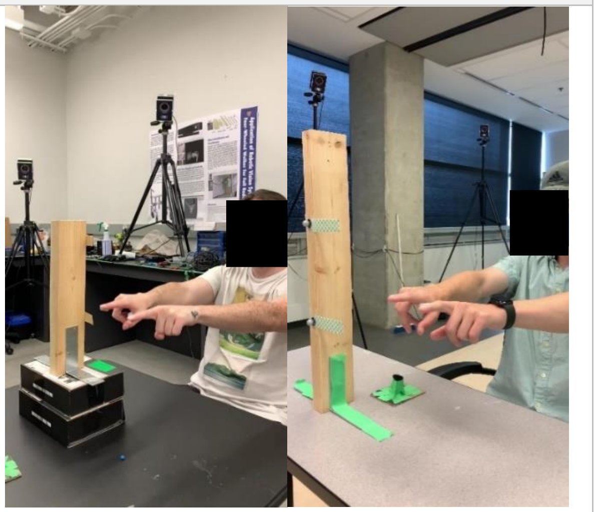
Figure 113: Top: Error in peak to peak distance between Mediapipe and Vicon. Bottom: Error in peak velocity between Mediapipe and Vicon

5.2 Appendix B.

Common Mistakes using the Computer Vision System for Finger Tracking

Table 20: List of common mistakes which lead to deteriorated tracking performance, with visual examples

Common Collection Issues	Visual Illustration
Camera set up too far away from the participant	
Incorrect perspective- camera is set up too low and pointing at an upward angle. Best performance is when camera is set up higher and is pointing down towards the hands.	 [redacted]

Incorrect angle- image on the left: Camera angle too close to the frontal view. Image on the right: Camera angle too close to the participant’s side view. Optimal angle is about 45 degrees from the sagittal plane, with no overlapping hands.	
Hands too close to each other	
Calibration board not placed/held firmly