DESIGN AND VALIDATION OF A SOFTWARE TOOL TO IMPROVE METHODS FOR CREATING CUSTOM ANALYSES OF MOTION CAPTURE DATA: ANALYSIS OF QUARTERBACK THROWING BIOMECHANICS BASED UPON A COACHING PERSPECTIVE

A Thesis

by

HUNTER WESTON STORACI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Michael R. Moreno
Committee Members,	Steven F. Crouse
	Michael L. Madigan
Head of Department,	Anthony Guiseppi-Elie

May 2017

Major Subject: Biomedical Engineering

ABSTRACT

The goal of this work was to develop a software platform to synchronize and facilitate the way in which researchers perform custom analysis of motion capture (MOCAP) data. The software was developed using Python and contains both frontend and backend tools to accomplish this goal. It includes a user interface which allows users to: access MOCAP trial data, view corresponding video files, display results of custom-programmed analyses, bulk analyze data, compile and save results, and generate PDF reports over the analysis. The backend contains a set of tools which: allow for easy access of MOCAP data stored in the C3D file format, establish a framework which future custom MOCAP analysis can follow, and efficiently stores results in a platform-independent, serialized, data file which can be opened at a later time for additional statistical processing of results.

To validate the software platform a human subjects research study was designed to analyze quarterback (QB) throwing biomechanics in a manner consistent with what football coaches are teaching to players. Interviews were conducted with eight coaches from the middle school and high school levels to establish the common "coaching points" being taught to players. A custom marker-set was designed for the study, and fifteen QBs (ages 12-18, mean 15.1) were brought in for data capture. Subjects were asked to perform three throws (hitch, corner, comeback), each using a different foot pattern (quick game, 3-step drop, rollout), to each side of the field. Throwing data was captured using a twelve-camera Vicon Motion Capture System (Vantage V16), and four AMTI force plates (OR6-6-1000).

Custom analyses were programmed to examine the coaching points from the interviews: elbow flexion, balance throughout the throwing motion, release time, release orientation, analysis of the stride, accuracy of the throw, and the hip leading angle were the points of interest. Additional analyses, from both the QB work and a separate MOCAP study, were programmed by undergraduate researchers in the lab, to verify the use of the software's usefulness to multiple users and applications.

ACKNOWLEDGEMENTS

To my parents, for their unwavering support in everything that I have done throughout my life. I owe it all to you. Thank you and I love you!

My advisor, Dr. Michael Moreno, thank you for all of the guidance over the years, allowing me to take on this project even though it was not what we originally agreed upon, and always being there for your grad students as we try to navigate life away from home.

All of the students in the BMEL: Aaron, Andrew, Brandon, Caleb, Lise, Mingliang, and Steve, thank you for accepting me into our weird little family and always being willing to discuss questions and provide insight and ideas. A big thanks to Andrew for the countless discussions on software and design, this project was just as much your idea as it was mine. To Andrea, Eric, and the Shannons, for helping on the long study days and nights.

To my siblings and friends, thank you for always listening and distracting me just enough to keep me sane. Kevin - you weren't the hero I deserved, you were the one I needed.

Thank you to the Physical Education and Activities Program at A&M for allowing us to use their facilities to run the football studies.

NOMENCLATURE

BMEL	Biomechanical Environments Laboratory
GUI	Graphical User Interface
MOCAP	Motion Capture
PEAP	Physical Education and Activities Program
SARA	Symmetrical Axis of Rotation Analysis
SCoRE	Symmetrical Center of Rotation Estimation

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis committee consisting of Professor Michael Moreno of the Mechanical Engineering Department, Professor Michael Madigan of the Biomedical Engineering Department, and Professor Stephen Crouse of the Department of Health and Kinesiology.

All work for this thesis was completed by the student in collaboration with Andrew Robbins of the Department of Biomedical Engineering, at Texas A&M University.

Funding Sources

There are no outside funding contributions to acknowledge related to the research and compilation of this document.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 Background and Motivation

1.1.1 Software Framework

Motion capture (MOCAP) systems are often used as a research tool for both motion optimization and rehabilitation settings, but analyzing the data outputs from a MOCAP system and presenting the data in a meaningful way has proven problematic. The majority of MOCAP systems come with a platform-specific piece of software that is used for building models, designing marker sets, performing analysis, and generating reports. There are also a number of commercially available and open source software packages that can be used to perform similar types of analysis. While easily accessible, these packages do not provide the flexibility needed for performing custom analysis and visualizations of the data for collaborators or in a manner that is most beneficial to a specific audience, be that scientific or non-scientific. This often results in researchers who utilize MOCAP technology writing their own custom software on a case-by-case basis to perform analysis and present data for each different MOCAP application. This requirement to consistently produce custom software results in large overhead for every project. As researchers move on from a lab, the custom code that they leave behind is often unusable by others due to a lack of commenting, platform dependencies, no clear way to execute it, etc. Thus, the next group of researchers are left with no choice but to design a new piece of software to perform similar operations, and the whole process repeats itself. There is a need for a platform-independent software that allows users to implement custom analyses and visualizations of MOCAP data and results in flexible ways for any audience, all while maintaining a common code architecture allowing for future use and expansion of custom code for different motion capture applications.

1.1.2 Quarterback Throwing Biomechanics

Throwing a football is a highly technical motion that requires precise coordination and body control to perform at a high level. Quarterbacks (QB) at all levels must master several techniques for throwing the ball, each to a high degree of skill and accuracy, to be successful. The technique employed depends on the route of the receiver; some throws require high release angle trajectories, while others require flat trajectories; certain throws may require high velocity, while others will require more "touch" on the pass. Other variables that are adjusted can include shoulder alignment, release point, posture, and footwork. Shorter routes require a quick release and a more economical arm motion while longer routes require more torque and lower body contribution to force generation. In addition, lower body concerns include, but are not limited to, throwing from the shotgun, a 1, 3, 5, or 7 step drop, or a designed "roll out" where the throw must be made on the run. Even the most basic pass requires highly technical coordination of body motion.

Beginning as early as 5 years of age, a child may be coached in the "proper" technique for throwing the ball. As the athlete progresses, they will continue to be instructed, by different coaches at different levels, on the proper technique. The technique being taught is likely based primarily upon qualitative assessments of empirical observations. Though coaches may agree on some general principles, there is no formal consensus on what constitutes "proper" technique. Moreover, there is little quantitative data to support the development of such a consensus or the acceptance of diversity as a basis for yielding to each individual's unique form. With the recent shift towards offenses at all levels becoming more reliant on passing and throwing, it is even more imperative that QBs are taught the proper mechanics at the youngest age possible.

1.2 Problem

1.2.1 Software Package

There are a number of open source and commercially available software packages that allow users to perform kinematic analysis, kinetic modeling, or musculoskeletal modeling of human motion. Yet researchers often work on applications that require custom analysis techniques or methods for visualizing the data. These existing software packages do not allow enough flexibility to perform these various analyses, to present the data in a novel way, or to present the data in a way that is useful for an audience who may not have expertise within the biomechanics field. This results in research labs writing new software each time they need to perform custom analysis for a different motion capture application. There is a large amount of overhead involved in developing custom software; researchers must learn new syntax and programming techniques, how to access and organize data outputs from a MOCAP system, or how to compile results into a format that is useful for statistical processing. Additionally, where software written by previous researchers should be a good starting point for a new student (especially for long term studies where multiple students are working on the same application), more often than not that software is only understood by the original programmer. There is a need for a software program that allows users to program application-specific analysis of motion capture data, gives them the flexibility to present and visualize the results for presentation to any audience, and maintains a common software framework that allows some measure of continuity to exist between the various MOCAP applications and researchers so that the code can be easily ran and understood by future researchers, even after the original developer has left the lab.

1.2.2 Quarterback Throwing Biomechanics

Relatively little biomechanical analysis has been performed with respect to the optimization of the quarterback throwing motion or towards the standardization of coaching techniques that are specific to individuals of a certain age group or skill level. Only two studies could be found that look at the kinetics and kinematics specific to the football throw. Rash & Shapiro (1995) videotaped 12 collegiate quarterbacks over 3 years at the Senior Bowl to investigate the kinetics and kinematics of the shoulder and elbow at the moment of foot contact [1]. Fleisig et al. (1996) used high-speed motion analysis to compare the kinematic and kinetic aspects of the football throw and baseball pitch using 26 high school and collegiate QBs and baseball pitchers [2]. An additional study, performed by Kelly et al. (2002), used electromyography to define the different phases of the football throw, but did no kinematic or kinetic analysis [3].

Each of these studies were limited in terms of either their subject set or the type of throws that the subjects were asked to perform. No analysis was done on middle school aged athletes, each study only looked at a single type of pass, and none made mention of trying to recreate game-like throwing scenarios. The analysis for each of these studies was done based upon what was previously seen in the literature for analysis of throwing mechanics and was geared for an audience of scientific peers. However, the groups which could most benefit from this analysis are the football coaches who are teaching the mechanics. For the data to be useful to these specific groups, it needs to be presented in a way that represents the "coaching points" that are being taught by the coaches themselves. These points are not represented in previous analysis of throwing mechanics, which limits the ability of a coach to use the data to make adjustments or determinations as to what constitutes proper throwing mechanics.

1.3 Specific Aims

1.3.1 Specific Aim 1

Determine a set of analyses of quarterback throwing biomechanics that is consistent with the most common "coaching points" that are being taught to football players at the middle school and high school levels.

1.3.2 Specific Aim 2

Develop a method of implementing marker based motion capture technology that will allow for accurate estimation of a subject's anatomical features and be usable with joint center estimation algorithms and various 3rd party software packages.

1.3.3 Specific Aim 3

Use the data from Aim 1 to design a human subject's research study that replicates game-like throwing scenarios and footwork patterns in a manner that is adaptable to data acquisition via motion capture and force plate technology and gather data for software validation purposes (see Aim 4).

1.3.4 Specific Aim 4

Develop and validate a custom software platform designed to be used in conjunction with motion capture and force plate data acquisition systems that (1) provides an intuitive graphical user interface for the researcher, (2) provides the ability to design custom analysis for acquired data, (3) provides the ability to view results in a variety of formats, including intuitive representations that can be easily interpreted by non-scientists (e.g. coaches and players), (4) provides scalability and adaptability for other motion capture applications, and (5) allows the user to compile and output results in a platform-independent format for downstream data processing.

1.4 Significance

The resulting software application will be expandable for any future analysis of QB throwing mechanics, flexible enough to allow users to use it as a tool when programming custom analyses for future motion capture studies, will establish a common programming framework so researchers can easily understand and re-use analysis performed by prior users, and will reduce the overhead required to perform custom MOCAP analysis. Additionally, the results from the QB study will be used to quantify a relatively understudied motion in a way that better replicates game-like scenarios than what has previously been seen. The results will be used to provide football coaches with analysis of the “coaching points” that they are teaching to their players, and to aide in the development of a formal consensus for what constitutes proper football throwing mechanics. The resulting software application will be used in year-over-year analysis of the progression of a quarterback’s mechanics, and will help us begin to determine what techniques should be taught to players based upon their age and skill-level.

1.5 Delimitations

The software was programmed using the Python Programming Language. This choice was made because Python is the language of choice for current researchers in the lab, was determined to be powerful enough to perform analysis of large data sets and to create a large scale software application, and utilizes simplistic enough syntax that even a future researcher with no programming background could learn it for MOCAP data analysis. The types of throws and foot patterns that were chosen for the QB study were based upon the results from interviews conducted with current middle school and high school football coaches. Subjects were asked to perform anywhere from 6-16 throws for the different throwing trials based upon when the lead researcher determined that enough trials with no marker set issues and clean foot strikes on the force plate had been captured. The focus of this thesis was on the development of the software framework, so the analysis performed in this was for a single subject, and is primarily used as validation of the capabilities of the software framework. Accuracy was determined based upon whether the subject accurately threw the football into a 4x4 foot area determined by a target. This area was determined after speaking to the football coaches about their training methods, and examining the targets that they use when working with their QBs.

1.6 Limitations

The first four participants performed the throwing trials with a non-optimal force plate arrangement, resulting in a struggle to get accurate force plate data on the longer throws; force plates were subsequently adjusted for later subjects. The subjects threw the football with 75 reflective markers attached to their skin. No subjects complained of discomfort, but there still may have been some adverse effects to their natural throwing motion. All subjects were given the option to keep their shoes on for the study, however some opted to throw barefoot; this may have resulted in an unnatural throwing motion. The subjects were throwing from a raised walkway inside a gymnasium. Although the throws were measured to replicate game-like throwing distances, there may have been some mental effect from throwing on a walkway or throwing indoors. Data capture sessions lasted anywhere from 1 ½ - 2 hours, including subject preparation time. No subjects complained of fatigue, but that potentially could have factored into performance. The order in which the subjects performed the throws was not randomized, it was the same for every subject. Subjects wore as little clothing as they were comfortable with, this meant that they all still wore some sort of spandex on their upper legs. Markers had to be affixed to the spandex which likely resulted in large motion artifacts for the markers on the thigh segment of the subjects.

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

The purpose is to develop a software tool that will allow for future researchers to easily analyze motion capture data and implement new methods for analyzing and visualizing motion capture data .The secondary purpose of this study was to determine the required analysis of the quarterback (QB) throwing motion that is most relevant to the coaches who are teaching the mechanics to players, then visualize the data in a way that clearly illustrates these "coaching points" and their application to coaching strategies. Therefore, this section will review the relevant literature pertaining to both of these study objectives.

2.2 Quarterback Throwing Biomechanics

2.2.1 Kinetic and Kinematic Analysis

The scope of the peer-reviewed literature for QB throwing biomechanics is narrow. To the authors knowledge, there are only two known peer-reviewed studies that look specifically at kinetics and kinematics of the football throwing motion, and all available studies are limited in terms of subject group, number of throws, or applicability of the study to game-like throwing situations.

Rash and Shapiro (1995) [1] utilized video tape of 12 collegiate QBs to investigate the kinetics and kinematics of the shoulder at foot contact. They performed their analysis with respect to the reference planes shown in Figure 1 and looked at the kinetics and kinematics of the throw at three discrete events that occurred after back foot contact.

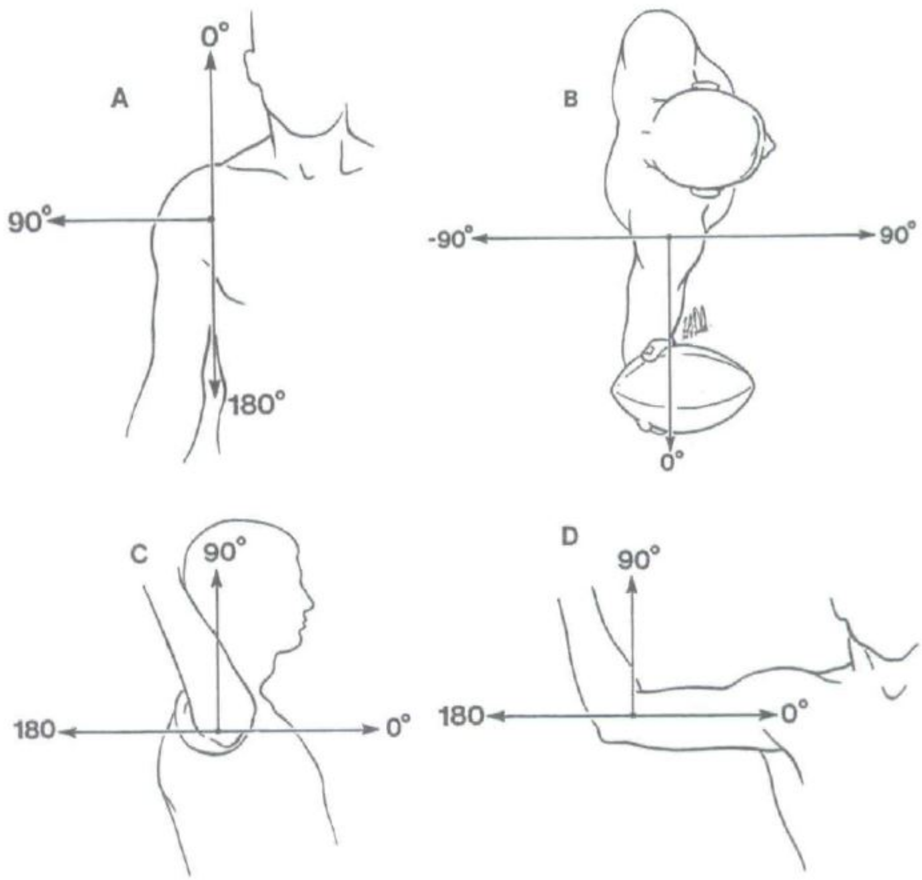


Figure 1: Reference values for kinematic and kinetic variables defined by Rash & Shapiro [1]: (a) shoulder abduction, (b) horizontal adduction, (c) external rotation, (d) elbow extension. Figure reprinted with permission from Human Kinetics, Inc.

Fleisig et al. (1996) [2] used high speed motion analysis techniques to compare the kinetics and kinematics of the football throw and baseball pitch using a subject set of 26 high school and collegiate QBs and pitchers. They analyzed the throwing motion with respect to the phases that had previously been used to describe the baseball pitch (Figure 2) and used the reference planes shown in Figure 3 and Figure 4 for their analysis.

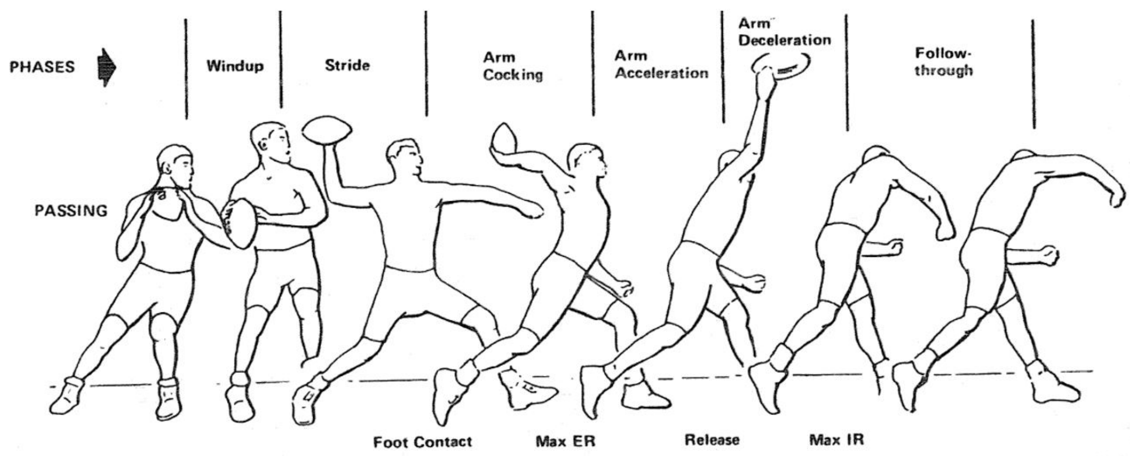


Figure 2: Using the phases of the baseball throw to define the football throw. As reported by Fleisig et al. [2]. Figure reprinted with permission from Human Kinetics, Inc.

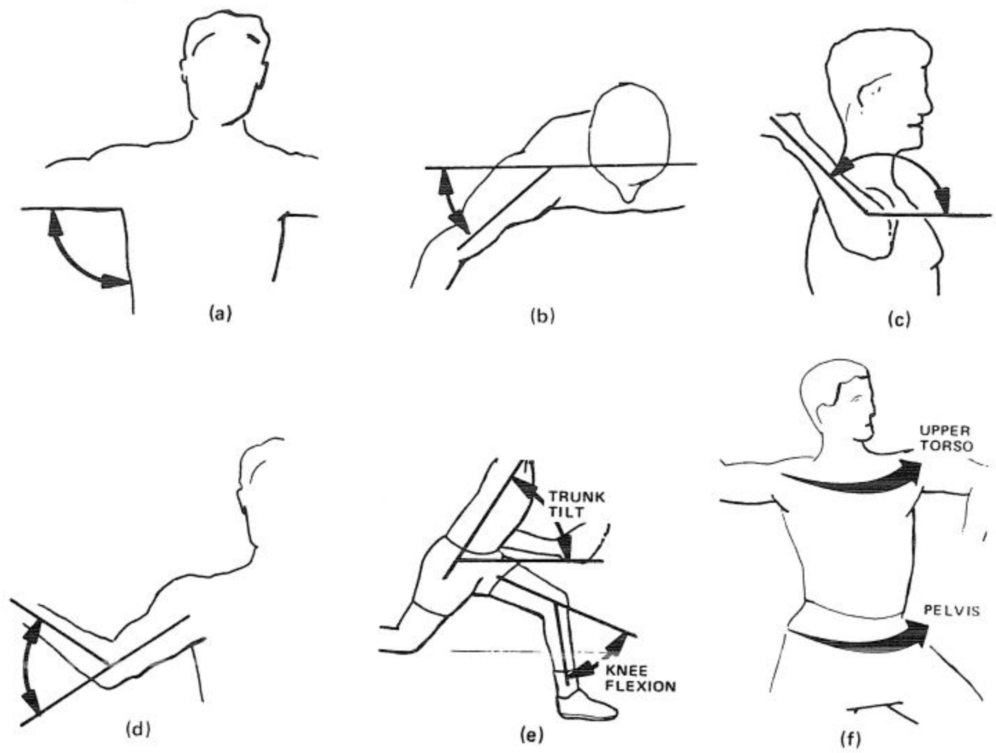


Figure 3: Kinematic variables as defined by Fleisig et al [2]: (a) shoulder abduction, (b) horizontal abduction, (c) external rotation, (d) elbow flexion, (e) lead knee flexion and trunk tilt, (f) pelvis angular velocity and upper torso angular velocity. Figure reprinted with permission from Human Kinetics Inc.

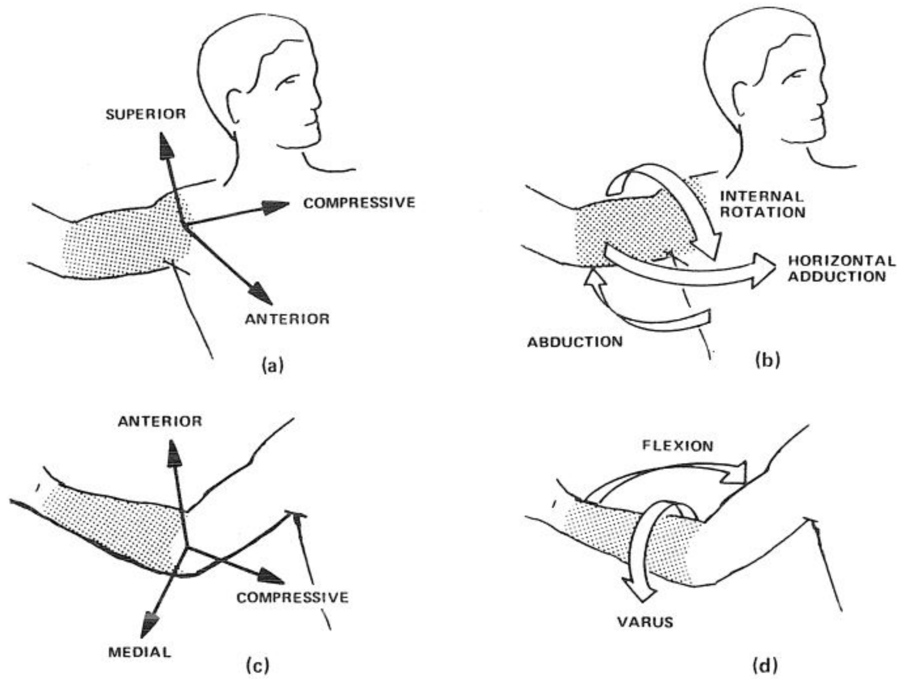


Figure 4: Kinetic variables reported by Fleisig et al [2]: (a) shoulder forces, (b) shoulder torques, (c) elbow forces, (d) elbow torques. Figure reprinted with permission from Human Kinetics Inc.

Although Rash & Shapiro and Fleisig et al. performed their analyses at different points in the throwing motion, there is enough similarity in the variables reported and timing of the analysis for some comparisons to be made. A comparison of the kinematic and kinetic parameters from the two studies is shown in Table 1 and Table 2.

Table 1: Kinematic parameters measured by Fleisig et al. [2] and Rash & Shapiro [1].

	Fleisig et al. (1996)	Rash & Shapiro (1995)
Instant of Foot Contact		
Stride length from ankle to ankle (% height)	61	-
Shoulder abduction (°)	96	83* (97)
Shoulder horizontal adduction (°)	7	-1
Shoulder external rotation (°)	90	47
Elbow flexion (°)	77	105* (75)
Lead knee flexion (°)	37	-
Arm Cocking Phase		
Max pelvis angular velocity (°/sec)	500	-
Max shoulder horizontal adduction (°)	32	-
Max upper torso angular velocity (°/sec)	950	-
Max elbow flexion (°)	113	-
Instant of Maximum Shoulder External Rotation		
Maximum shoulder external rotation (°)	164	164
Maximum shoulder external rotation velocity (°/sec)	-	2987
Elbow flexion (°)	-	95* (85)
Elbow flexion velocity (°/sec)	-	1276
Shoulder horizontal adduction (°)	-	7
Shoulder horizontal adduction velocity (°/sec)	-	851
Shoulder abduction (°)	-	96* (84)
Shoulder abduction velocity (°/sec)	-	1725
Arm Acceleration Phase		
Maximum elbow extension velocity (°/sec)	1760	-
Average shoulder abduction during acceleration (°)	108	-
Instant of Ball Release		
Ball velocity (m/sec)	21	18
Shoulder external rotation (°)		136
Shoulder external rotation velocity (°/sec)	-	1063
Shoulder abduction (°)	-	96* (84)
Shoulder abduction velocity (°/sec)	-	4
Shoulder horizontal adduction (°)	26	12
Shoulder horizontal adduction velocity (°/sec)	-	154
Elbow flexion (°)	36	59* (121)
Elbow flexion velocity (°/sec)	-	1225
Trunk tilt forward (°)	65	-
Trunk tilt sideways (°)	116	-

Table 1 Continued

	Fleisig et al. (1996)	Rash & Shapiro (1995)
Lead knee flexion (°)	28	-
Arm Deceleration Phase		
Maximum shoulder internal rotation velocity (°/sec)	4950	-
Maximum elbow flexion (°)	24	-
Average upper torso angular velocity (°/sec)	310	-

***Variable defined in terms of the reference plane defined by Fleisig et al.**

Table 2: Kinetic parameters measured by Fleisig et al. [2] and Rash and Shapiro [1].

	Fleisig et al. (1996)	Rash & Shapiro (1995)
Arm Cocking Phase		
Maximum shoulder anterior force (N)	350	-
Maximum shoulder horizontal adduction torque (N*M)	78	-
Maximum shoulder internal rotational torque (N*M)	54	-
Maximum elbow medial force (N)	280	-
Maximum elbow varus torque (N*M)	54	-
Instant of Maximum Shoulder External Rotation		
Shoulder distraction force (N)	-	-435
Shoulder distraction force (N)	-	250
Shoulder anterior force (N)	-	233
Shoulder superior force (N)	-	-66
External rotation torque (N*m)	-	-60
Adduction torque (N*m)	-	-889
Horizontal adduction torque	-	-345
Elbow distraction force (N)	-	236
Elbow medial force (N)	-	162
Elbow anterior force (N)	-	-47
Medial deviation torque (N*M)	-	69
Arm Acceleration Phase		
Maximum elbow flexion torque (N*M)	41	-

Table 2 Continued

	Fleisig et al. (1996)	Rash & Shapiro (1995)
Instant of Ball Release		
Shoulder distraction force (N)	-	-320
Shoulder anterior force (N)	-	20
Shoulder superior force (N)	-	-150
External rotation torque (N*m)	-	-19
Adduction torque (N*m)	-	18
Horizontal adduction torque	-	23
Elbow distraction force (N)	-	-242
Elbow medial force (N)	-	16
Elbow anterior force (N)	-	129
Extension torque (N*m)	-	-11
Medial deviation torque (N*m)	-	23
Arm Deceleration Phase		
Maximum shoulder compressive force (N)	660	-
Maximum elbow compressive force (N)	620	-
Maximum shoulder adduction torque (N*M)	58	-
Follow-through Phase		
Maximum shoulder posterior force (N)	240	-
Maximum shoulder horizontal abduction torque (N)	80	-

The purpose of the current work is not to provide a complete kinetic and kinematic analysis of the football throwing motion, but rather to analyze the aspects of the throw that were deemed most important for coaching analysis and visualize the results in a way which would maximize the benefits to coaches and players. Therefore, only the relevant kinematic variables measured in the two studies are discussed.

Fleisig et al. reported the stride length of the throwers to be ~61% of their height. Elbow flexion and shoulder rotation at the moment of foot contact were defined by Fleisig et al. as 77° and 90° and by Rash & Shapiro as 105° and 47° respectively, with the maximum elbow flexion reported as 113° during the cocking phase by Fleisig et al.. The maximum shoulder external rotation was described by both studies as 164°, with Rash & Shapiro reporting a 95° elbow flexion at this same instant. At the instant of ball release, Rash & Shapiro reported an elbow flexion and shoulder external rotation of 136° and 59° while Fleisig et al. reports an elbow flexion of 36°. Fleisig et al. reported a ball velocity of 21 m/s while Rash & Shapiro reported this value as 18 m/s. During the arm deceleration phase Fleisig et al. reported the elbow reached a maximum flexion of 24°.

To compare the studies it is useful to briefly discuss the methodology. Fleisig et al. looked at 26 high school and college quarterbacks, and had each quarterback perform a drop back and throw a pass that was equivalent to the length of a baseball pitch (18.4m). They utilized the Motion Analysis Corporation Expertvision 3-D software to reconstruct the locations of 13 surface markers to estimate the joint centers used for kinematic and kinetic analysis. Rash & Shapiro used 12 college quarterbacks, and had them throw a 30 yard pass off of a drop back. The researchers used two 60Hz video cameras, and the Peak Performance Video Analysis System to digitize the video records for each of the subjects.

It is difficult to compare all of the kinetic and kinematic parameters between the studies due to the fact that in many instances the parameters were recorded at different points during the throwing motion. However, a general comparison can be gained by comparing certain parameters. Similar values (within the reported error) were found for shoulder abduction (108° vs 96°) during the arm acceleration phase, and horizontal shoulder adduction (26° vs 12°) at the point of ball release. Fleisig et al. found significantly higher values for some maximum angular velocities in the throwing arm (elbow flexion – 1760 vs 1276°/sec, and internal rotation – 4950 vs 2987°/sec) at similar points in the throwing motion. Overall, Fleisig et al. reported larger values for many of the kinetic parameters as well: 54 N*m vs 19 N*m for elbow varus torque, 660N vs 435N for shoulder compressive force, and 620N vs 345N for elbow compressive force. Fleisig et al. cited a difference in data sampling rates as a potential reason for the large discrepancies between the studies. Rash & Shapiro used 60Hz cameras to capture the throwing motion, and due to the high speed motion of the throwing arm the low sampling rate may have not been sufficient.

2.2.2 Electromyography Analysis

Kelly et al. (2002) [3] used electromyography (EMG) techniques to define the different phases of the football throw and show how they differ from those of the baseball pitch, but performed no kinematic or kinetic analysis of the throwing motion. This is the only known study to define the phases of the football throwing motion in different terms than have been previously used to describe the baseball pitch. The definition of the throwing motion is shown in Figure 5.

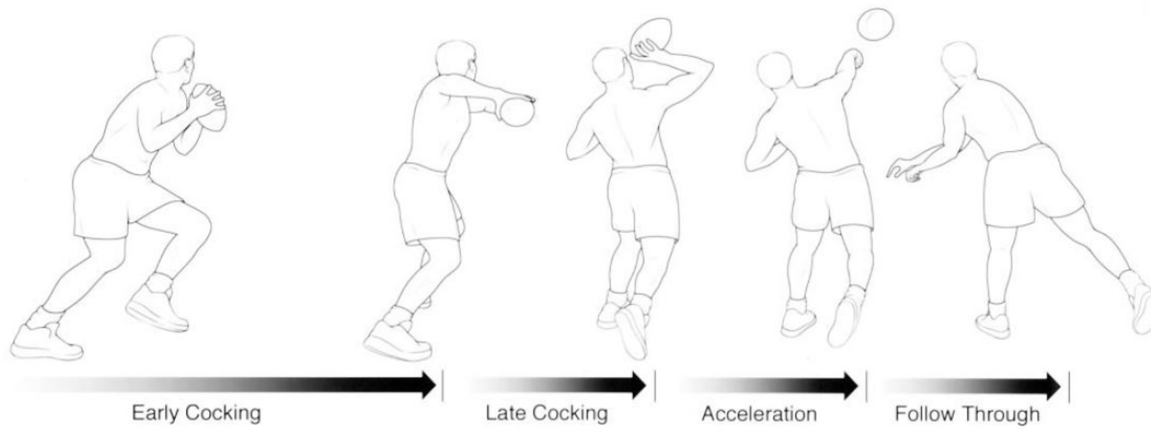


Figure 5: Using EMG to define the phases specific to the football throw. As reported by Kelly et al. [3]. Figure reprinted with permission from Human Kinetics, Inc.

2.2.3 Additional Sources

It should be noted that while only the studies listed above could be found in the peer reviewed literature, five graduate/undergraduate theses were found that performed some variation of analysis for the football throwing motion. As they are not peer reviewed, the methods and analysis for each of the studies cannot be verified, so only a brief summary of each is given.

Robert Heppe (1992) [4] used the Peak Performance 3-D Motion Analysis System and video cameras to investigate the kinematic variables related to the efficiency of the football throw. Four division 1-A quarterbacks performed a straight throw out of a 5-step drop back. The goal of the study was to determine the relationship between foot placement and accuracy of the throw. No correlation was found between the two parameters.

Jeremy Wood (2000) [5] used a Peak 5 2D Motion system and a Tekscan measurement system to investigate the footwork of high school QB's and relate it to their passing performance. The goal of the study was to relate the passing accuracy and velocity of the ball to the selected footwork patterns. It was found that the QB should step to the left of the target but within 10 degrees of the line to achieve maximum accuracy and velocity.

Brian Platt (2012) [6] used electromagnetic sensors to analyze the kinematics of the motion at four different phases of foot contact, specifically in high school athletes who played both the football QB and baseball pitcher positions. Significant differences were found in the degree of elbow flexion and the velocity of hip rotation at the moment of foot contact, as well as a difference in elbow flexion at the instant of maximal external rotation.

Anthony Beeman (2015) [7] used the Denavit-Hartenberg Method combined with a planar two bar mechanism to model the kinematic systems in the arm during the throwing motion. He then used Abaqus to create a Finite Element Model (FEM) to perform kinematic analysis based upon the model of the overhead throwing motion. The goal was to better model the kinematics of the football throw to determine optimum kinematics for the QB throwing motion.

Kyle Bohnert (2016) [8] used 11 motion capture cameras, Cortex software (Motion Analysis Corp), two force platforms (Bertec Corp), and electromyography (Delsys) to perform kinematic, kinetic, and EMG analysis of the football throw. Three collegiate quarterbacks were examined in an attempt to determine the proper mechanics involved in the football throwing motion.

2.3 Motion Analysis Software

2.3.1 C3D File Format

The C3D file format provides the ability to store 3D and analog data in an unprocessed form and make it intelligible to any user because of the file format's standard properties [9]. In addition to storing 3D and analog data, C3D files can contain information on the physical design of the laboratory (analog channels, force plate orientation, etc.), trial information (date, sample rates, etc.), patient information, as well as any calculated results from models that are being used [9]. The use of the C3D file format is widely adopted in the biomechanics field, and allows for sharing and accessing data without a dependency upon the hardware with which it was captured.

2.3.2 Kinematic Analysis and Modeling

Visual3D

Visual3D is a commercial software from C-motion [10] that is used for measuring movement and force data collected from a wide variety of motion capture systems. Visual3D allows for you to use traditional models or develop your own custom models using your own marker set, allows for kinematic and inverse dynamic calculations, and allows for you to create custom reports of your analysis [11]. A license for the Visual3D software costs $10,995 for 4 licenses, and additional licenses can be purchased for $2000 each.

Vicon BodyBuilder & Polygon

BodyBuilder and Polygon are the proprietary software platforms that are meant to be used with a user's Vicon Motion Capture System [12]. BodyBuilder allows for users to create custom models, or to utilize a library of pre-built models. Polygon is a reporting and presentation tool that can be used with a Vicon system to create custom reports. An advantage of using these software platforms is that they work in real time with a Vicon system. Disadvantages are that they cannot be used with any other motion capture system.

BTK, Mokka, and Open MA

While the C3D file format is widely used in the motion capture field, there are a limited number of software packages that allow a user to access and modify the data contained in a C3D file. To deal with this issue Dr. Arnaud Barr and Dr. Stephane Armand developed the Biomechanical ToolKit (BTK) to be used as an open-source and multiplatform framework for reading, writing, modifying, and visualizing data from any motion analysis system [13]. The development of this software stopped in 2014, and Dr. Barre has begun work on developing a more robust open source platform (OpenMA) that will allow for access to the C3D data as well as processing, analysis, and modeling of motion capture data [14]. However, this software is still under development and not yet available for use in the motion capture community. Mokka is a tool developed along with BTK and

OpenMA that allows for easy visualization and some simple analysis of motion capture trajectories and analog data [15].

BodyMech & the HuMAnS Toolbox

BodyMech is a MATLAB based open source package for kinematic analysis that was developed at the VU University Medical Center in Amsterdam [16]. However, this software was last updated in 2006 and was never widely adopted by the field.

The HuMAnS toolbox was primarily developed for research in humanoid robotics but also contains a biomechanical model of the human body and proposed a set of tools for the analysis and simulation of human and humanoid movement. However, there is very little documentation on the software and the last known edits were in 2009. Again, this software was never widely adopted by the field.

2.3.3 Musculoskeletal Modeling

OpenSim (SimTK)

OpenSim is a free, open source software platform that lets users develop models of musculoskeletal structures and simulate the dynamics of movement [17]. In addition to musculoskeletal modeling, OpenSim also allows users to provide inverse kinematic and kinetic analysis on their generated models [18]. OpenSim is being developed as a part of the larger SimTK project.

Biomechanics of Bodies (BoB)

BoB is a biomechanical modeling package containing a human musculoskeletal model that enables various calculations such as anatomical trajectories, center of mass, and ground reaction forces [19]. The package resides in MATLAB.

AnyBody

AnyBody is a commercial software package that is used for musculoskeletal modeling [20]. The package allows users to import data directly from a motion capture system to an AnyBody model and use those to calculate a number of parameters such as muscle forces, joint forces, or elastic energy in tendons.

2.4 Discussion

This section explored the current state of the literature with regards to the kinetic and kinematic analysis of the QB throwing motion. The limitations of the QB literature are not only that there are very few sources of verified information available, but also that the studies that have previously been performed were geared towards an audience of scientific peers and not the coaches and players who may have an interest in making use of the data. Previous studies were also limited in terms of their subject group, types of throws performed, or efforts to relate the study conditions to game-like scenarios. There is a clear need for a study that takes into account these different variables, as well as reports the data in ways that benefits an audience of coaches and players who can then use the data to improve upon their techniques.

Also discussed were the current software tools available to researchers for performing analysis of motion capture data. While there are a number of commercially available packages, it can be seen that these packages are very expensive or don't allow for the necessary flexibility to perform custom data analysis specific to any motion capture study. Another key element of using motion capture for scientific research is the visualization and presentation of data in unique ways, which may help non-experts to better understand and qualify the results of the study. There is currently no software package that allows for creating these types of visualizations in the same program where the analysis is performed; meaning that any researcher wanting to create these types of visualizations must always use a secondary software or must write their own software to create the visual representations of the data. The proposed software packages will allow for the capabilities of performing both custom data analysis and visualizations, all within the same framework.

CHAPTER III

HUMAN SUBJECTS RESEARCH STUDY DESIGN

3.1 Introduction

To design a research study that would allow for the most robust and useful analysis of the QB throwing motion, we first needed to identify the parameters that would be used for analysis. This was accomplished through interviews with four middle school and four high school football coaches. The goal was to determine the common "coaching points" being taught at each level of play, as well as the common foot patterns used by QBs and routes run by receivers during games. These data were then used to design a human subject's research study that incorporated "game-like" throwing scenarios and analysis that has not previously been seen with regards to the QB throwing motion. The data could then be presented in a way that is representative of what coaches are actually looking for. Allowing them to use the results as a coaching tool for training of quarterbacks at the different levels. Additionally, the data will be the first time point in a longitudinal study aimed at quantifying a relatively understudied motion in age groups not previously seen, and looking at the progression of the throwing mechanics as a QB progresses through years of training..

3.2 Interviews with Coaches

The coaches interviewed had experience ranging from 1- 31 years (mean - 16.25). The goal of the interviews was to summarize all of the responses from the different coaches and compile a list of the most common techniques that are being taught and the most common throwing routes that are used at the different levels. For a technique/route/foot pattern to be chosen for analysis, we determined that it must be repeated by at least 3 of the coaches who were interviewed and it must not be directly contradicted by one of the interviewed coaches.

IRB approval (IRB2016-0211D) was obtained from the Texas A&M Institutional Review Board prior to the initiation of the interviews.

3.2.1 Questions

In each interview, after recording some basic information (name, coaching position, institution, and experience as a coach), the same series of questions was asked to determine the following data: the most common throwing routes and foot patterns used by their players in game scenarios and what they are coaching to their QBs from a mechanics perspective.

3.2.2 Interview Results

The most common foot patterns utilized by QBs on the interviewed coaches' teams were:

- No drop back (i.e., throwing out of the shotgun)
- 3-step drop
- 7-step rollout

The most common receiver routes seen in games for each of these teams were:

- Quick game (i.e., screen, slant)

- Post/flag route
- Curl/dig/hitch route

After taking into account differences in terminology used among the different coaches, a total of 46 different coaching techniques were recorded from the interviews. Of these techniques, 11 of them were mentioned in at least half of the interviews. Based upon the 11 points that matched our criteria, the following points were chosen for the analysis of the throwing motion:

- Hip leading angle
- Elbow leading the hand
- Release time
- Orientation at ball release
- Consistency of the throwing motion
- Off-hand motion
- Accuracy
- Stride length and direction

3.3 Marker Set Design

A new marker set, Quarterback Template, was created for use in the football study. The main goals set prior to the creation of the marker set were to allow for compatibility with SCoRE and SARA algorithms, to achieve flexibility for performing analysis based upon surface or virtual markers, and to find a balance between capturing accurate data (more markers) and not limiting the subject’s natural throwing motion (less markers).

Marker cluster designs were based on an article by Cappozzo et al. (1997) [21] and from the documentation provided for Visual 3D [10]. These guidelines state that four markers per segment is ideal with marker placement selected to minimize motion artifact, the long axis of the cluster should be along the long axis of the segment, and that clusters should be as non-collinear and widely distributed on the segment as possible.

The only segments in the model that did not have four markers were the hands, feet, and shoulders. This was because we could not reasonably fit four markers on a hand without obstructing the subject, the subjects wore shoes so there was no need to put extra markers on the feet, and there was not an obvious 4th marker location for the shoulders due to the large motion artifact from the muscles and bones located there. The final result is shown in Figure 6, and a subject affixed with the marker set is shown in Figure 7:

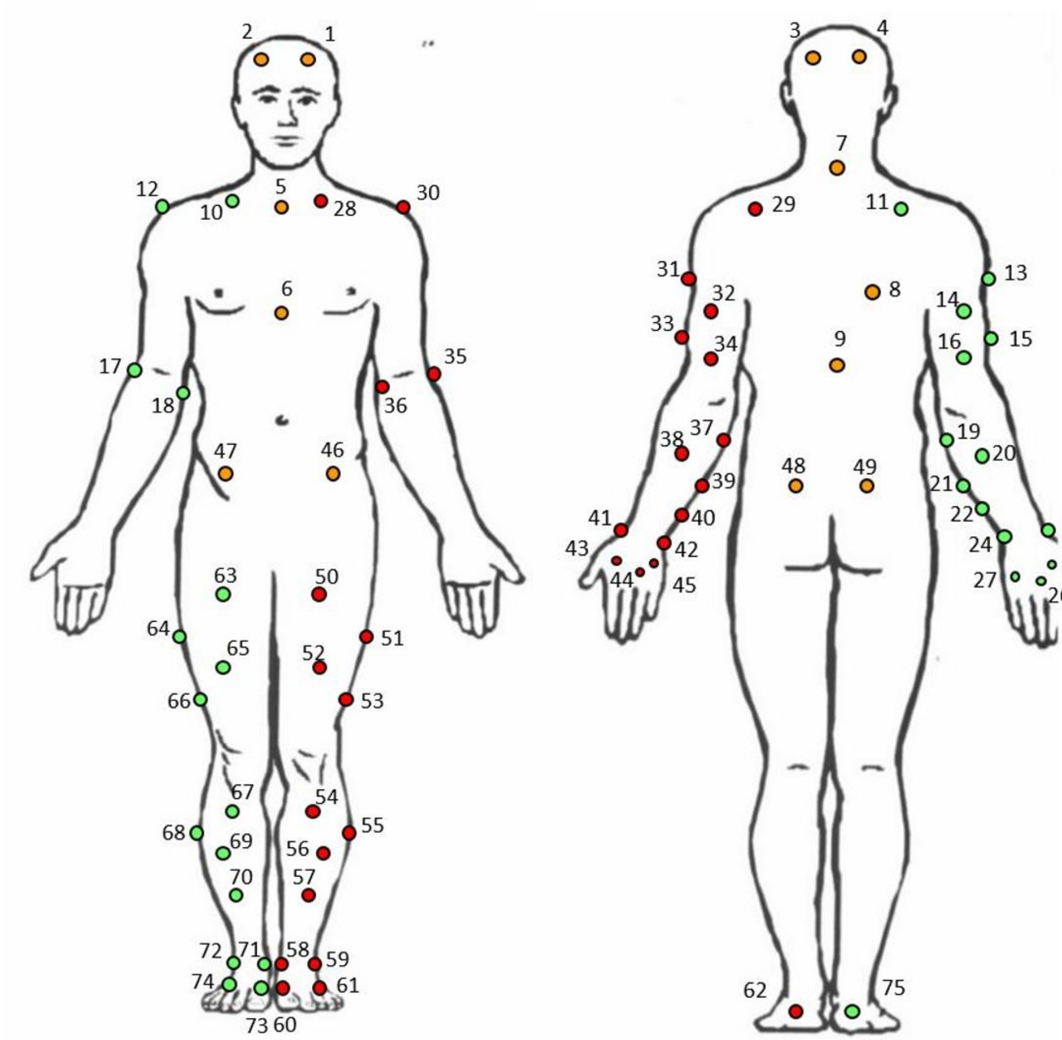


Figure 6: Quarterback template marker set used in the study.

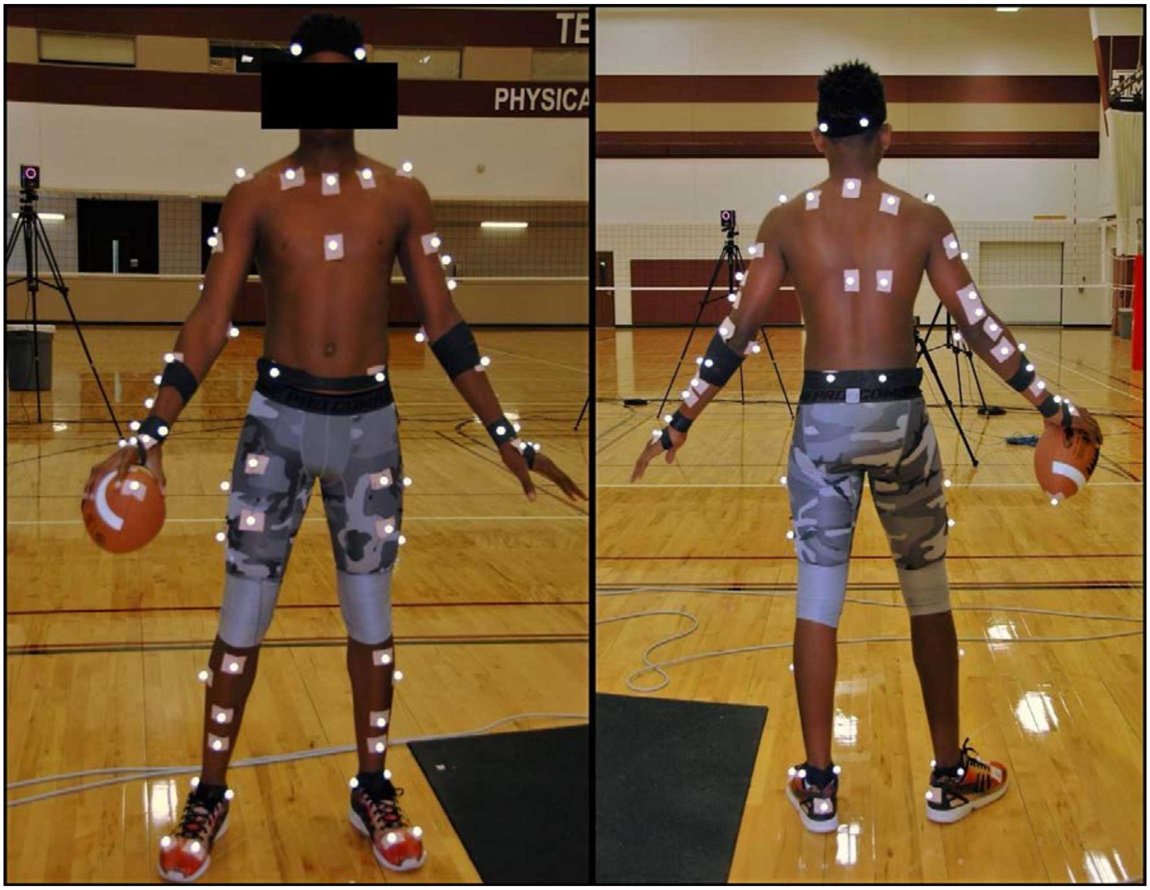


Figure 7: QB with reflective markers attached.

3.4 Research Study Design

Based upon the input from the coach interviews, a research study was designed to accurately replicate the throwing scenarios that a QB is most likely to see in a game situation. Three different throws, each with a different foot pattern, were determined to be sufficient for this purpose. Although, due to coaches expressing differences seen when throwing to opposite sides of the field, each subject was asked to throw each pattern to both sides of the capture volume, resulting in 6 different throws, utilizing 3 different foot patterns.

3.4.1 Participants

Fifteen QBs were brought in for data capture (ages 12-18, mean 15.1), 6 middle school (6^{th} (1), 7^{th} (3), 8^{th} (2)) and 9 high school (9^{th} (3), 10^{th} (3), 12^{th} (3)). The middle school or high school classification was based upon the players highest year of competition completed as a QB (i.e., an incoming freshman was considered a middle schooler). The QBs in the study were all current middle school or high school players and were referred by their current high school football coach.

IRB approval (IRB2016-0290D) was obtained from the Texas A&M Institutional Review Board, prior to the start of this study.

Consent was obtained from all subjects (parental consent, and minor assent for all subjects under the age of 18) prior to a subjects participation in the study. Additionally, all subjects were asked to fill out a brief health questionnaire to ensure there were no major health restrictions that could affect their participation in the study. Data capture took place in a large gymnasium located at the Texas A&M Physical Education Activities Program (PEAP) Facility that allowed for subjects to perform common game-like throwing scenarios to targets placed at similar distances to what they would see when throwing on a football field.

3.4.2 Equipment

A twelve-camera Vicon Motion Capture System (Vantage 16MP), with six wide angle, and six narrow angle lens cameras, was used to obtain 3D marker data for the throwing trials. Four AMTI Force Plates (OR6-1000) were used to obtain force data for the throws that occurred on the raised platform. Three Bonita Video Cameras (Vicon 720p) were used to obtain video data for the each of the subjects throws. Two video cameras were placed within the capture volume and the 3rd was placed behind the target to record the accuracy of the throws. Cameras were arranged as shown in Figure 8. Significant gaps were left in the front right and front left of the capture volume to protect the cameras from the football and to give the subjects a clear line of sight to the throwing targets. Calibration was determined to be adequate when all cameras a calibration error of no greater than 0.2 as reported by Vicon Nexus.

3.4.3 Marker Attachment

On the study day, each player was marked up according to the Quarterback Template marker set (Figure 6) developed in the BMEL for use in this study. Subjects wore only compression shorts or leggings during the trials. Subjects used either a high school or middle school regulation football and were given the option to throw with or without their shoes on. Moleskin was used to attach the markers and bases to the subjects skin. Self-stick tape was used on the subjects' hands, wrists, ankles, and any other area where there was difficulty getting the markers to stay stuck in place. Safety pins were used to secure moleskin patches to shoes and clothing in an effort to limit the inaccuracies due to motion artifact from not having markers placed directly on the subject's skin. All subjects were told to inform the study director if anything felt uncomfortable prior to or during the study so as to not interfere with their natural throwing motion.

For consistency, markers were placed by the same three trained technicians for each subject in the study. Technicians were taught how to palpate the bony landmarks (i.e., medial epicondyle or lateral malleolus) prior to the start of the study, and the lead researchers checked all marker placements prior to the start of throwing trials to ensure correct placements. For marker clusters, technicians followed the guidelines set forth by Cappozzo et al. [21] and C-Motion [11].

3.4.4 Throwing Trials

Each subject was given time to warm up his throwing arm in the time between the calibration trials while researchers worked to manually label the marker template. Subjects threw three different throwing patterns, each with a different foot pattern, to both sides of the field, resulting in six different throws and 3 different foot patterns being used in the study. The routes thrown (in order) were:

- 5 yard hitch to the right, out of the shotgun (i.e., just catch and throw)
- 25 yard corner to the right, out of a 3-step drop back
- 5 yard hitch out to the left, out of the shotgun (i.e., just catch and throw)
- 25 yard corner to the left, out of a 3-step drop

- 12 yard curl to the right, out of a rollout
- 12 yard curl to the left, off of a rollout

The first four throwing patterns were thrown from a raised platform with four embedded force plates. The subjects threw anywhere from 6-16 throws for the various patterns until the lead researcher was confident that adequate force plate data (clean foot strikes for the front and back foot) were obtained. Due to the arrangement of the force plates within the walkway, the subjects starting location on the walkway was adjusted from one throw to the next to ensure optimal foot strikes for both the front back foot during the throwing motion. The rollouts were performed off of the walkway so no force plate data were obtained during these trials. For this reason subjects were only asked to perform 6-10 throws for these routes.

3.4.5 Camera and Target Locations

A 4x4 foot target was placed at each of the target locations. A vertical target was used for the hitch and comeback routes, as these are patterns that are generally thrown with a higher velocity on a much straighter trajectory, and a drop net was used for the corner routes, as this is a route that is thrown with a more arching trajectory. An image, and diagram of the set-up for the study is shown in Figure 8 and Figure 9.

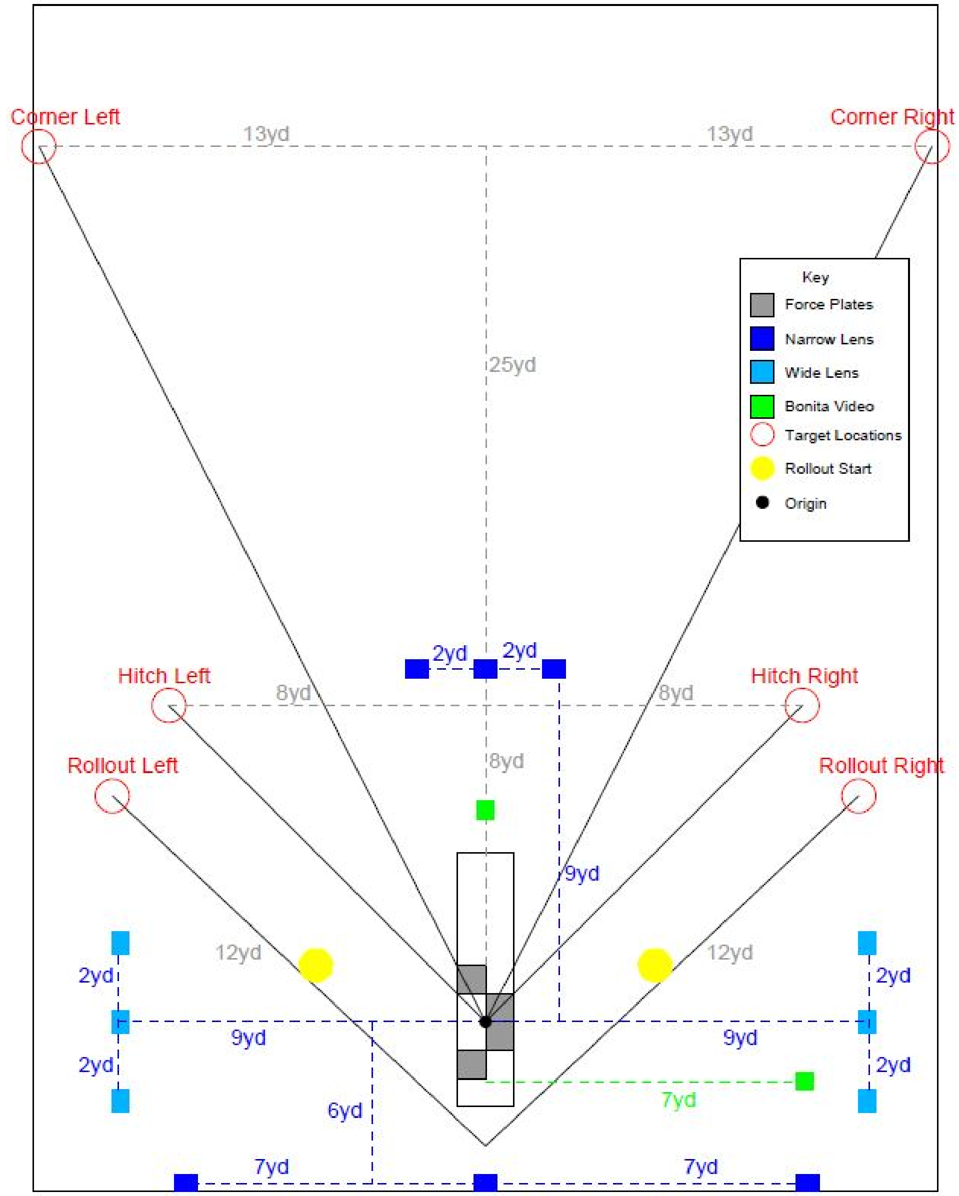


Figure 8: Camera, force plate, and target locations for the motion capture study.

Figure 9: Camera set-up at the Texas A&M PEAP facility.

3.5 Discussion

This chapter discussed the methods that were used to determine the relevant parameters for analysis and design a marker set and motion capture study that best captured these parameters. The primary goal of the motion capture study was to make it as "game-like" as possible, meaning having the subjects perform throwing trials to targets at similar distances to common throws that are seen in games while also using foot patterns that they are likely to use in a game scenario. Due to the high variety that is seen in football offenses and the different capabilities of quarterbacks at the varying levels designing a study that was representative of all teams was difficult. The best way to do this was determined to be to interview a variety of coaches, both from the middle school and high school level, and compile a list of the most common routes, foot patterns, and coaching methods that are used by each of their offenses. The most common results determined from the interviews could then be used to design a study that best represents what is seen and expected from all quarterbacks.

A new marker set, Quarterback Template (Figure 6), was designed for use in this study. The marker set was designed to be as flexible as possible in terms of its use with traditional motion capture modeling techniques (i.e., Vicon PlugInGait), Vicon SCoRE and SARA virtual joint modeling techniques, and 3rd-party software platforms for musculoskeletal modeling (i.e., OpenSim).

Data capture took place in a gymnasium that allowed for targets to be placed at distances similar to game-like throwing situations. The twelve Vicon cameras were arranged in order to maintain an optimal capture volume, while both protecting the cameras from thrown footballs and allowing the humans subjects space to move around in the capture volume without feeling obstructed by the cameras and cables. The optimal camera arrangement was determined to be with the wide angle lenses on the sides of the volume, where they could be placed closer to the subject, and the narrow angle lenses on the front, back, and corners. The final arrangement can be seen in Figure 8.

CHAPTER IV

SOFTWARE PACKAGE

4.1 Introduction

The software package was programmed using the Python programming language. Python is a high-level, open-source, dynamic programming language that is commonly used in scientific fields [22]. The goal of this project was to limit the amount of frontend work required to implement custom analyses of MOCAP data, and establish a framework that future programmers can follow when implementing these analyses. This was done by creating a simple framework for creating project-specific analysis scripts in Python and allowing a user to execute them from within a custom user interface (UI). The user can create their own custom analysis script and add it to a global analysis subfolder which contains all of the various analysis scripts that have been created by users of the interface. They can then use the UI to associate their script with the specific MOCAP project it was written for. From this point, the UI can be used to select MOCAP data files for a specific project, run the analysis scripts on the data for that trial, and view the results in the UI. From the main interface, the user can also create a new project if working on a new MOCAP application, interact with the video files for a specific trial, bulk process data, compile and output results, or create PDF reports detailing the results from the analysis. The software platform can be used to assist in the development of custom analysis for motion capture data returned in the c3d file format.

4.2 Python Modules

This next section describes the various Python modules that were used for creating the UI and the different tools that are required to operate the UI and program custom analyses of MOCAP data.

4.2.1 NumPy

NumPy is a common scientific computing package used with Python [23]. It allows users access to an N-dimensional array object used to perform fast computations on large data sets. These computations include linear algebra calculations, trigonometric math, array slicing operations, statistical operations, etc. NumPy arrays are similar to Python list objects but operate more efficiently to improve the speed of the Python program. NumPy is used in nearly every Python file that was written to create the final software package

4.2.2 Matplotlib

Matplotlib is a plotting library that can be used to create interactive figures within a Python script [24]. It is written to be compatible with NumPy and has many useful features that allow users to embed plots with graphical user interfaces (GUI). Matplotlib is used to create the visualizations of the results from running an analysis, and is also used to display the results using interactive plots inside the main window.

4.2.3 PyQt, Qt, and Qt Designer

PyQt is a set of Python bindings for the Qt application framework [25]. Qt is a set of C++ libraries for platform-independent creation of application software that can run on a wide variety of software and hardware platforms without having to change the underlying codebase [26]. The main window, all of the different UI features, and the majority of the popup windows that can be accessed from the main window were first created using Qt Designer [27], and then compiled into Python files using PyQt.

4.2.4 ReportLab

ReportLab PDF Library is a software that allows users to directly create documents in Adobe's Portable Document Format (PDF) using Python [28]. This library helps automate the process of generating PDF reports while still allowing for the generation of professional looking PDF documents. ReportLab is used in the software to create custom PDF reports over the analyses that are run on a MOCAP project.

4.2.5 PyMuPDF

PyMuPDF is a Python binding of the MuPDF libraries which is a lightweight PDF viewing software. The main advantages of MuPDF are speed and its capabilities for opening and rendering PDFs for viewing within the user interface of the software platform [29].

4.2.6 C3D

The c3d library is meant for reading and writing c3d files [30]. Upon installation, this library was not fully compatible with c3d files returned from the Vicon Motion Capture System, and some minor edits were made in the source code to fix these compatibility issues; however the bulk of the library remained in place. To access the required information, an additional Python module was created that makes use of some of the features in the c3d modules to access the specific c3d file information that is required for creating analysis scripts over MOCAP data.

4.2.7 Pickle

Data storage is performed using Python's object serialization protocol: Pickle. "Pickling" is the process by which an object is converted to a byte stream that can then be stored into memory. "Unpickling" is the reverse process in which the byte stream is converted back into its object hierarchy [31]. This process is used to store both project and trial specific information after creating new projects or running analyses so that the data can be accessed at a later time without having to redo the work, or so that the data can be shared between researchers or accessed on different computers that do not have all of the source code that was originally used to create the file.

4.2.8 Pandas

Pandas is a Python module providing fast, flexible, and expressive data structures designed to make working "relational" or "labeled" data both easy and intuitive [32]. Pandas provides a number of advantages for working with data and results:

- Easily handles missing data
- Can store data of any type
- Can insert and delete columns from data, as well merge existing data sets
- Intelligent slicing and indexing of data sets

For all of these reasons listed we chose to output the final results from using the software into a Pandas DataFrame. This should ensure a seamless transfer from the results outputted by the software package, to a user being able to perform downstream processing (statistics, comparisons, etc.).

4.3 Software Design

4.3.1 Overview

The final package consists of a number of project-independent classes that are meant to assist in performing custom analysis of MOCAP data. The final class set consists of:

- Classes for the main UI and the various popup windows and features that can be accessed from it
- Classes for long-term data storage
- Classes for the global data analysis features and for parsing the c3d data file
- Class for performing basic math operations on the motion capture trajectories

- Classes written as plugins to add additional functionality to the main UI

Additionally, analysis specific to the QB mechanics project and validation of the software package was done by writing a set of individual analysis classes and one parent analysis class. A diagram outlining the class structure and interactions for the final package is shown in Figure 10.

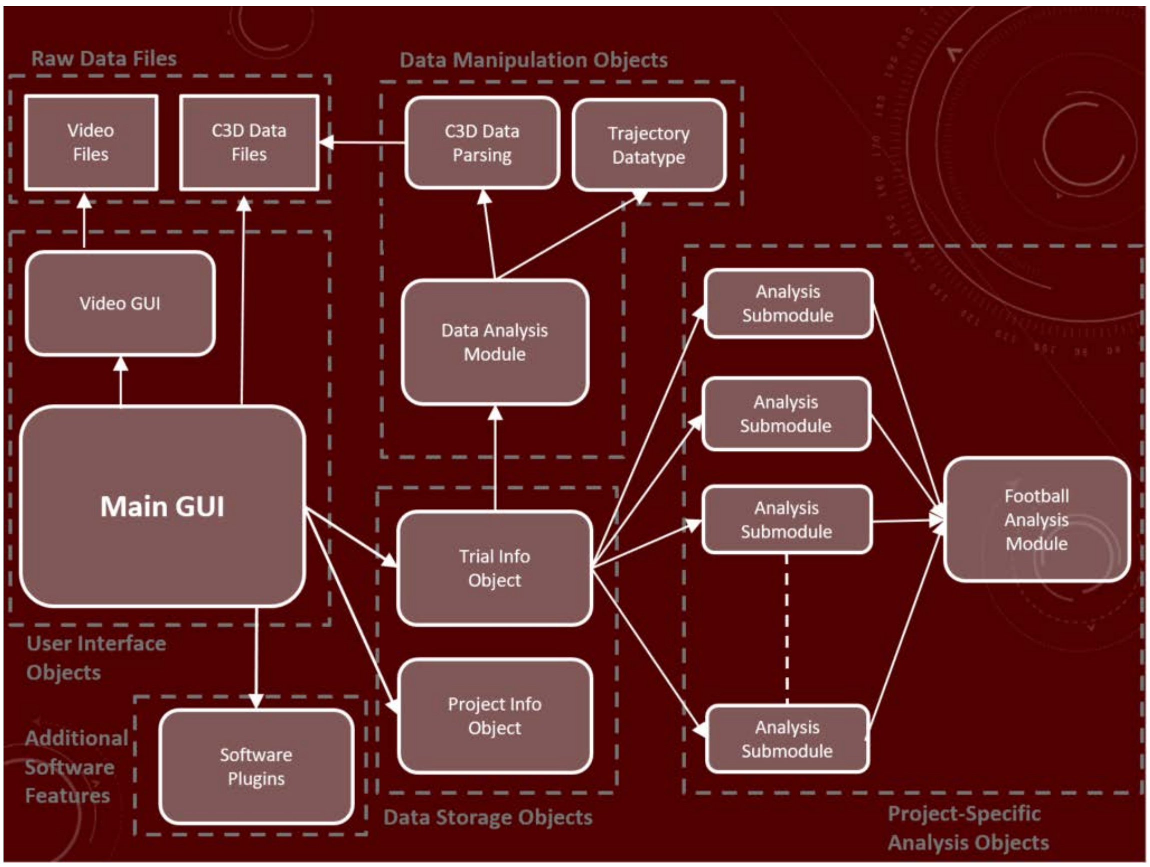


Figure 10: Design overview of classes for the final software package.

4.3.2 User Interface

Main Window

Overview

The main window consists of a current project display, an error display, a file tree display, and a trial information display, all of which are shown at all times. There are analysis and reporting tabs which the user can navigate between and buttons that allow the user to change the current project, view video files, run the current set of analyses, or select the active subset of analyses (for scenarios when they only need to run certain analysis scripts). The main window contains a menu bar with options for adding and removing projects, adding or removing the analysis scripts that are associated with a given project, editing the metadata that is associated with a given trial or project, entering into a "developer mode", and accessing the different plugins that have been added to the UI. A view of the main window is shown in Figure 11.

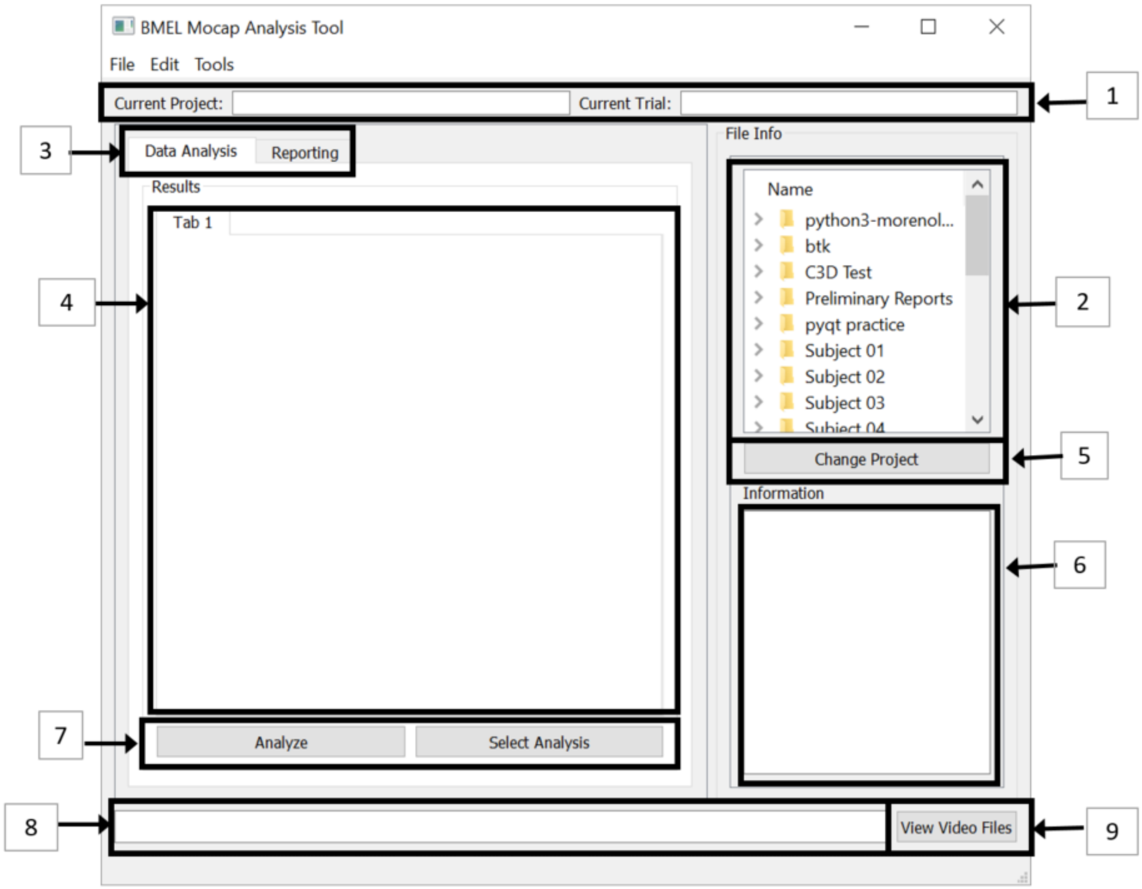


Figure 11: View of the main window interface with main features highlighted: 1) Current project and trial view, 2) File tree to navigate to different subjects and trials, 3) Tab widgets to switch between analysis view and reporting view, 4) Main viewing window for results, 5) Button for user to switch the current working project, 6) Display for information about the current trial, 7) Buttons to initiate analysis or change subset of working analyses, 8) Error display, 9) Button to open media viewer.

Adding and Removing Projects

The list of projects is dynamically loaded from a specified file each time the main window is loaded. When a researcher wants to use the software platform to run analysis for a new MOCAP project, they will first have to add that project to the list of current projects using the “Add Project” option in the edit menu in the main window. The new project will then be saved to the list of available projects. This popup window is shown in Figure 12.

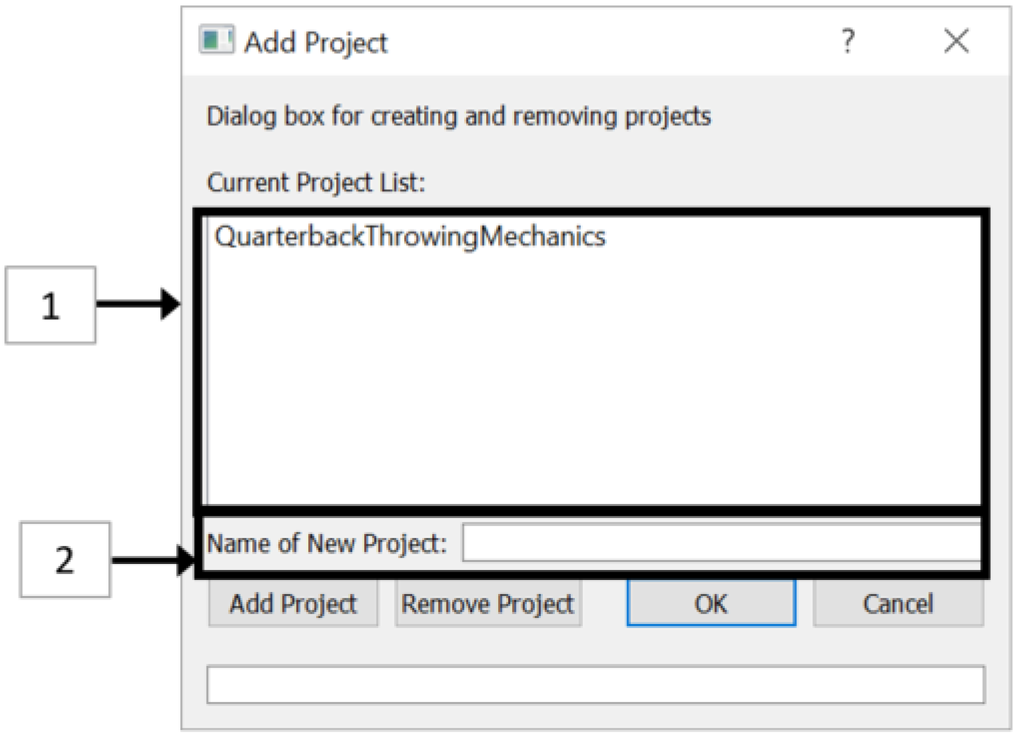


Figure 12: Example of how a user can add a new project for future motion capture applications: 1) A dynamically loaded list of all of the existing projects, 2) The user can type in the name of the new project and then add it to the list of existing projects.

Adding and Removing Analyses

As future researchers create their own custom analysis scripts for various MOCAP applications, these files will be added to a global analysis subfolder within the file architecture. The user then will have to use the “Add Analysis” option from the main window edit menu to associate any new analysis script with the correct MOCAP project.

This has to be done because when the user clicks the “analyze” button in the main window, only the analysis files that are associated the current project are executed. This process for the Quarterback Throwing Mechanics project is shown in Figure 13.

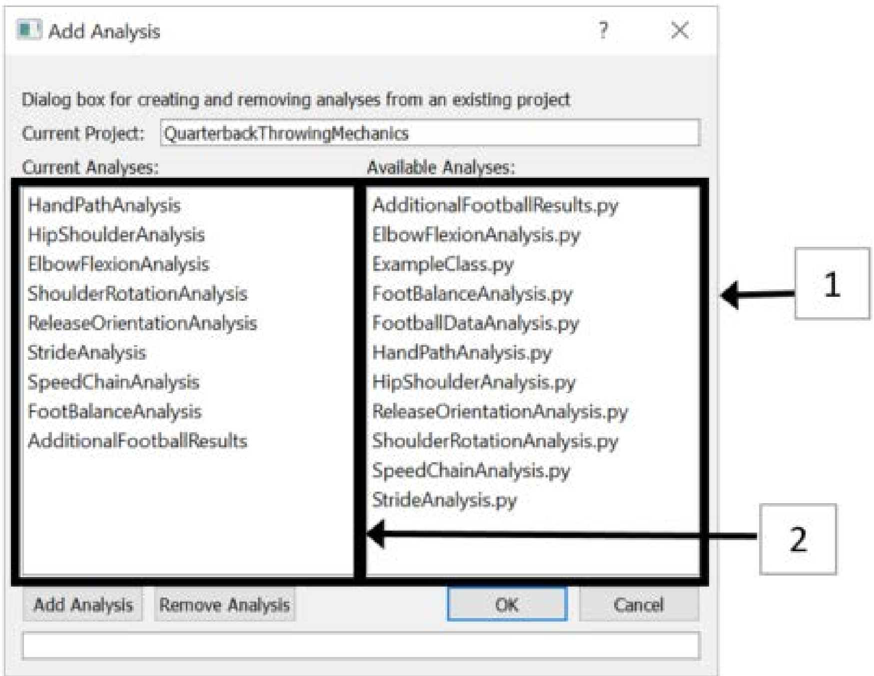


Figure 13: An example of how a user can add or remove analyses from an existing project: 1) A dynamically loaded list of all the available Python scripts located within the global analysis subfolder, 2) The analyses that are currently associated with the active project. The user can add or remove any of the available analyses from this list.

Adding and Editing Metadata

Also within the main window, the user has the option to edit the metadata associated with the current project. The reason for having a subset of metadata for each project is to allow users provide additional information about a project or subject that may not be the result of a calculation but that is necessary to differentiate between trials, organize the data statistical analysis, or to perform additional analyses outside of the UI. For the QB study, this metadata may include things like a subject's age or weight, the school that they attend, the grade they are in, or their experience as a quarterback. An example of potential metadata needed for an animal study would be the specific breed of an animal, and whether it was male or female. Figure 14 gives an example of how a user can add metadata options to a MOCAP project, and Figure 15 shows how a user can edit the metadata values for specific MOCAP trials.

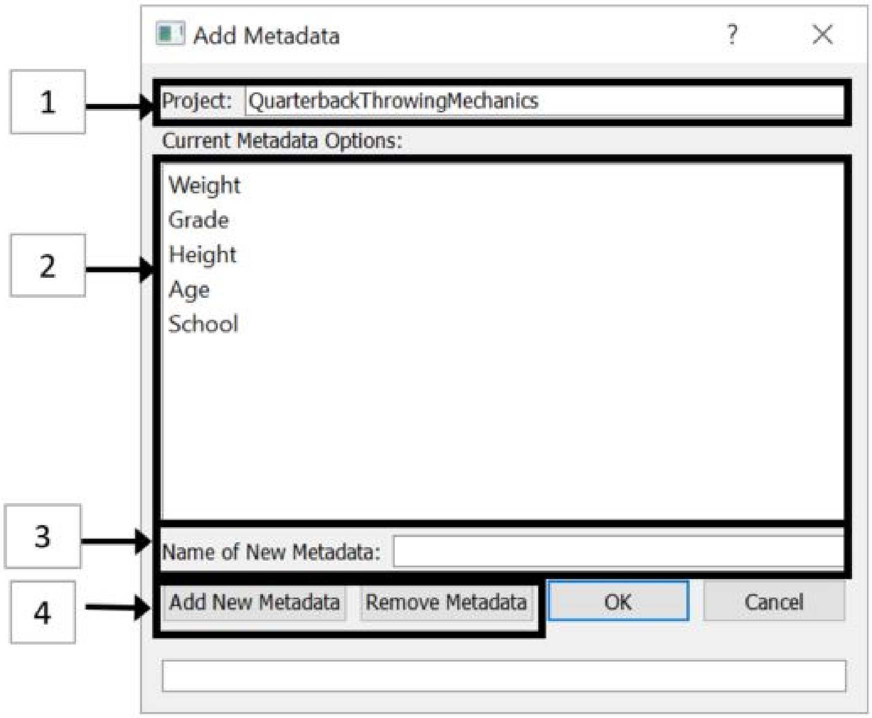


Figure 14: Example of how a user can add metadata options to a given project: 1) The current project, 2) The current metadata associated with that project, 3) The name of a new metadata option that the user wants to add to the current project, 4) Buttons for the user to add or remove metadata for a project.

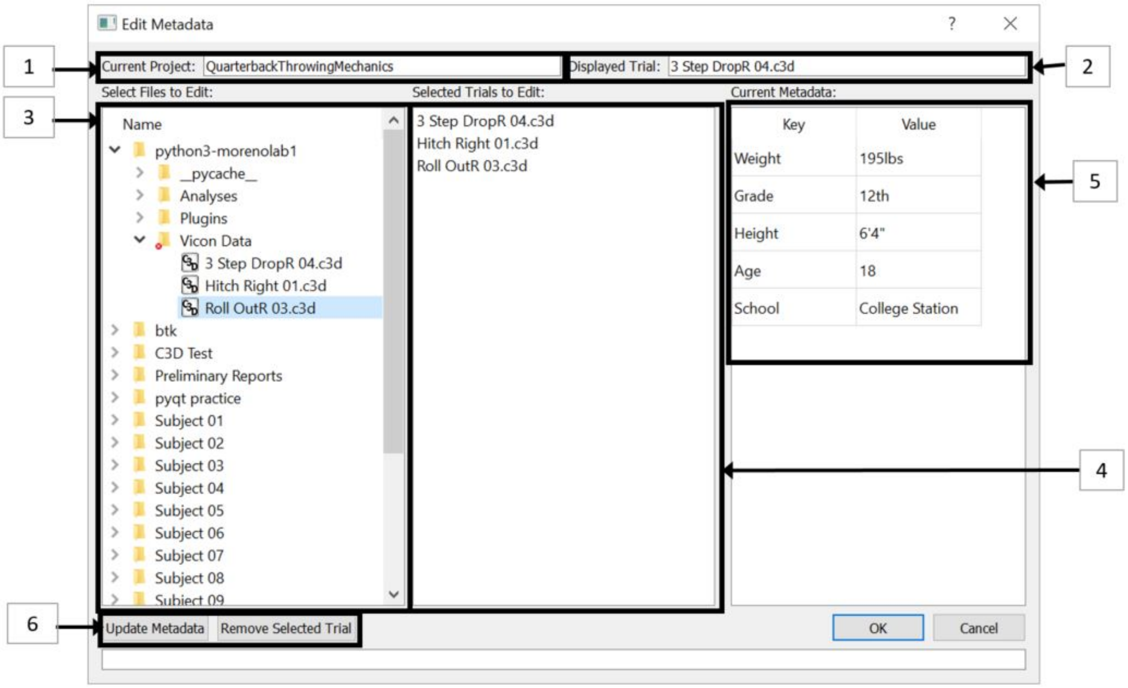


Figure 15: Example of the edit metadata window for the Quarterback Biomechanics project: 1) Display of the current project, 2) Display of the trial whose metadata is shown in (5), 3) A file tree where the user can select the trials for which they want to edit metadata for, 4) Display of the currently selected files, 5) Display of the current metadata for the given trial, this is where the user can edit the values, 6) buttons that the user can click to either update the metadata for each selected trial, or remove a trial from the selected trials list.

Analysis Tab

The analysis tab portion of the main window is where the results from running the analysis scripts are displayed in the UI. These results are displayed using at feature of the Qt library called a QTabWidget [33]. A tab is added for each result that is returned from running the analyses and the title of each tab is labeled the same as the name of the analysis class that returned it. This allows for a user to run any number of analyses return the results and still be able to display them within the UI. One of the major advantages of this is the ability to return matplotlib figures from the analysis scripts and display them within the QTabWidget with an interactive toolbar that allows for panning, zooming, and moving the plot to alter their view of the results. An example of this is shown in Figure 16.

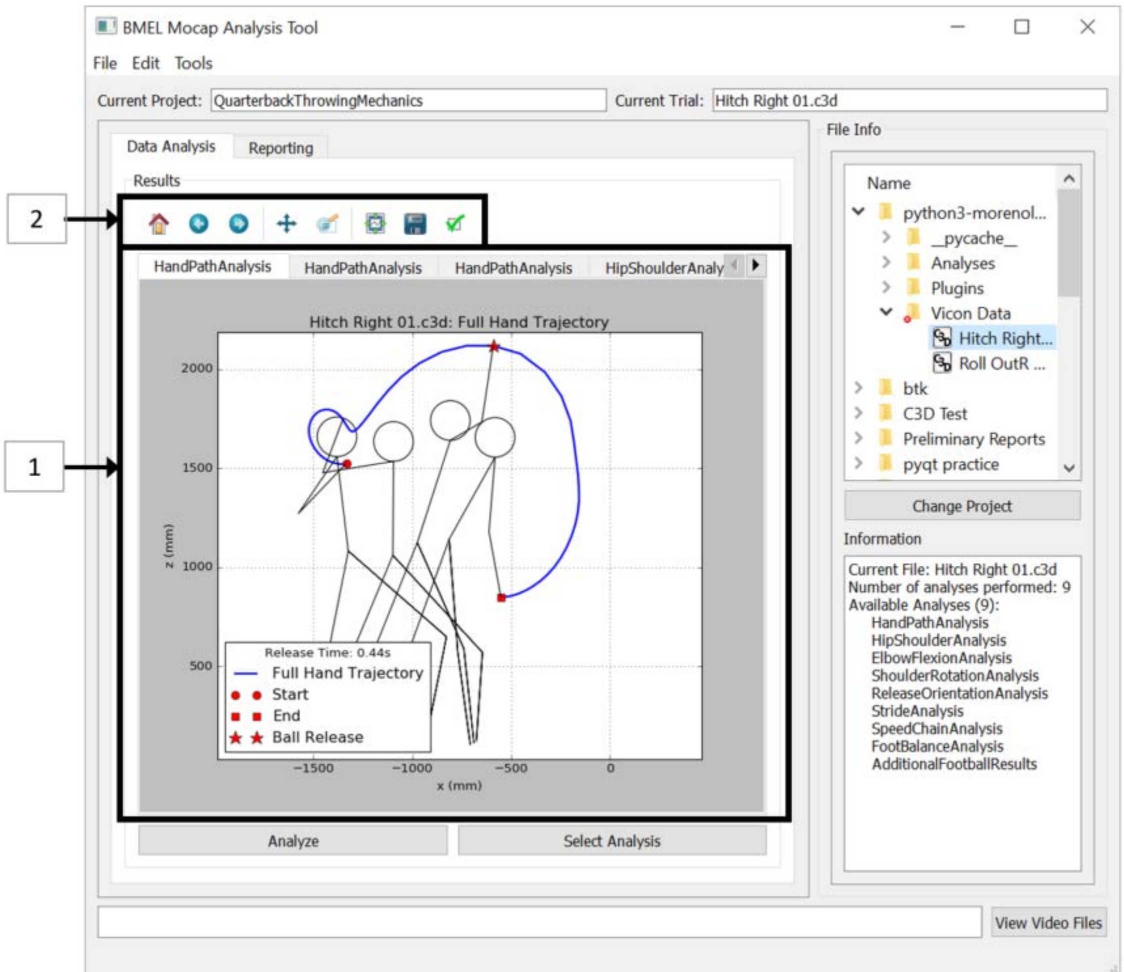


Figure 16: Main window view after analyses have been performed on a trial: 1) View of the results, and tabs for the user to navigate between the different analysis results, 2) Interactive toolbar for the user to pan, zoom, adjust axes, etc. on the results plot.

Reporting Tab

The reporting tab is the section of the main window where the user can create and display PDF reports over the analyses that have been run on a C3D trial. The different pages of the PDF output are displayed in a QTabWidget with the label of each tab corresponding to the analysis result that is displayed there. The user can also make edits to the cover page of the report within this interface. There is functionality to change titles, captions, and the images displayed on the cover page. The user has options to generate a report with or without a cover page, with each report type slightly altering how the results are displayed within the PDF output file. Users also have the option to change the output file name or save location of the final report. The layout of the reporting tab is shown in Figure 17.

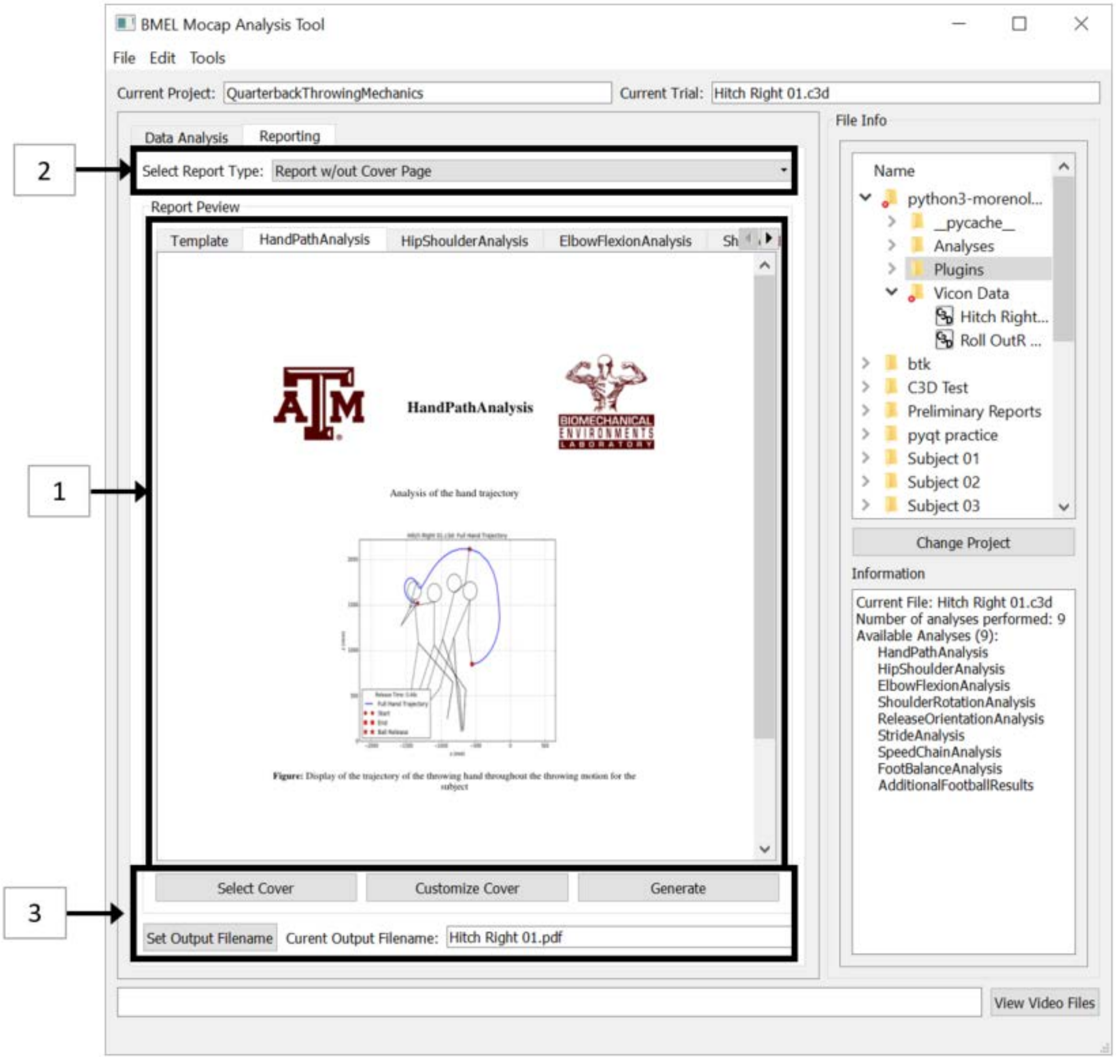


Figure 17: View of the main window with the report generation features highlighted: 1) Preview of the each of the individual PDF pages for the report, 2) User-selected report type, 3) Options for the user to customize certain aspects of the report.

Development Mode

Another feature of the UI is the option for the user to enter into “development mode.” When the user selects this option, the layout of the UI changes, and the majority of the UI features are disabled so that the user is essentially confined to the main window. When this option is selected and the run analysis button is clicked, whichever analysis modules that the user currently has selected are dynamically re-loaded before they are run. This feature allows the user to make edits to an analysis script then run the updated script from the UI and view the updated results without ever having to close the main window. This feature allows for interactive editing of analysis scripts while the UI is active, and reduces the amount of work required to update analysis files that are stored in memory. This is a useful feature for when a user is attempting to implement a new analysis for a project, and is likely to be making a lot of updates to the Python file. A display of the UI while in "development mode" is show in Figure 18.

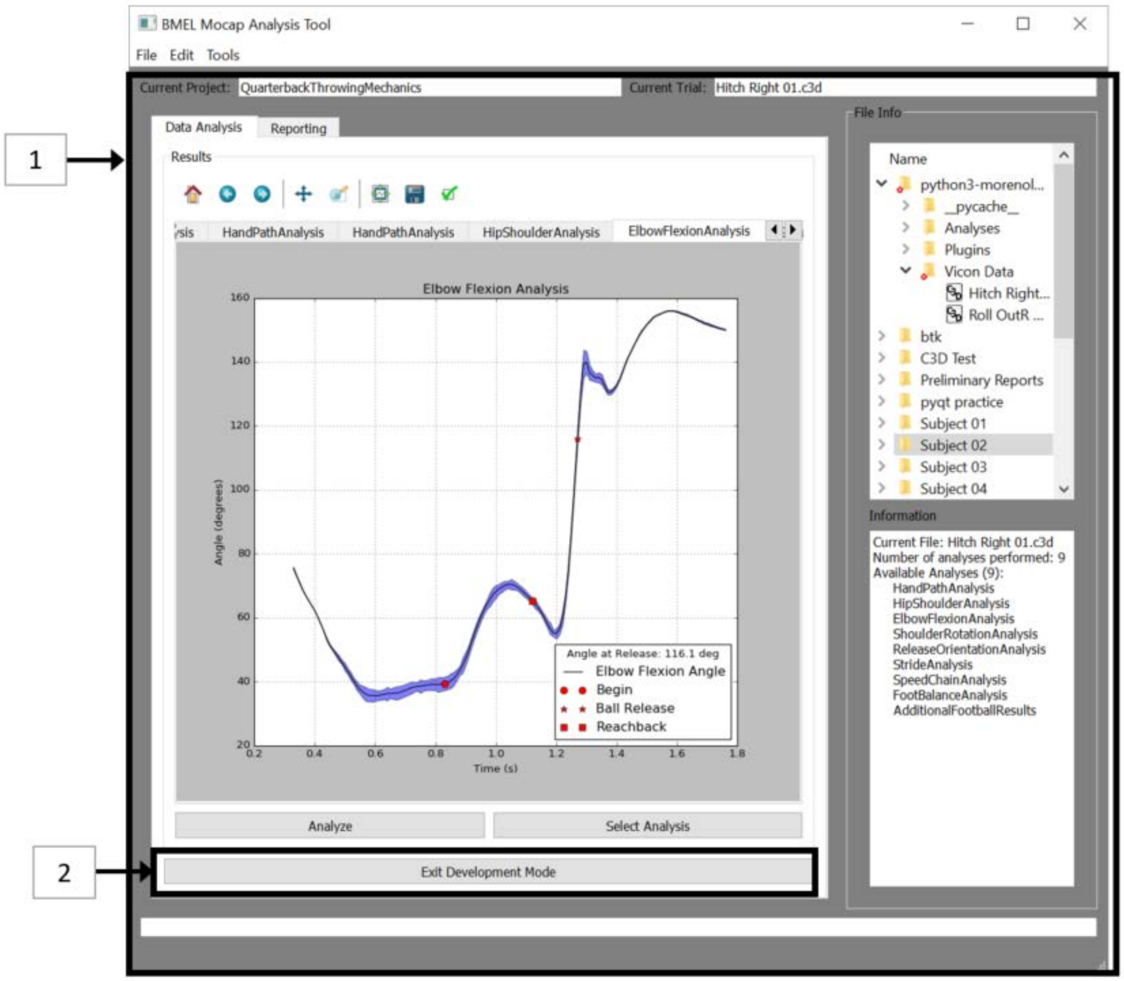


Figure 18: Image of the main UI while in development mode: 1) The interface changes background color when in this mode, selected analysis files will be reloaded prior to running so that updates can be shown in the results display, 2) Button for the user to exit development mode and enter back in to the normal window viewing.

Media Viewer

The media viewer popup window is displayed when the user clicks on the view video button in the main UI. From this window the user has the option to view all of the videos found in the file architecture with the same name as the active trial. There are options for viewing the video at different speeds, changing brightness and contrast settings, and viewing in full screen. The main purpose of the window, however, is setting the times when the “key events” for the trial occurred. These events are defined by the programmer in each of the different analysis classes. For example, one of the football analyses scripts involves looking at the orientation of the QB at the instant of ball release. In order to perform the analysis, the script needs the time when the ball was released. So inside the “StrideAnalysis” class, the programmer defines a list of events and gives one of them the name “release.” This list of events is loaded into memory when the user instantiates a trial displayed in the media viewer window. The user can then see the required events, and manually set the video frame in the trial when the specified event occurred. This is done by clicking on the “set events” button, selecting the event from the list, scrolling to the correct frame in the video, and then clicking on the “set time” button. At this point, the current video frame is associated with that event, and the updated time is communicated back to the main UI and saved in a dictionary of event times which is then passed as a parameter to the different analysis scripts when they are run. An image of the video viewer window is shown in Figure 19.

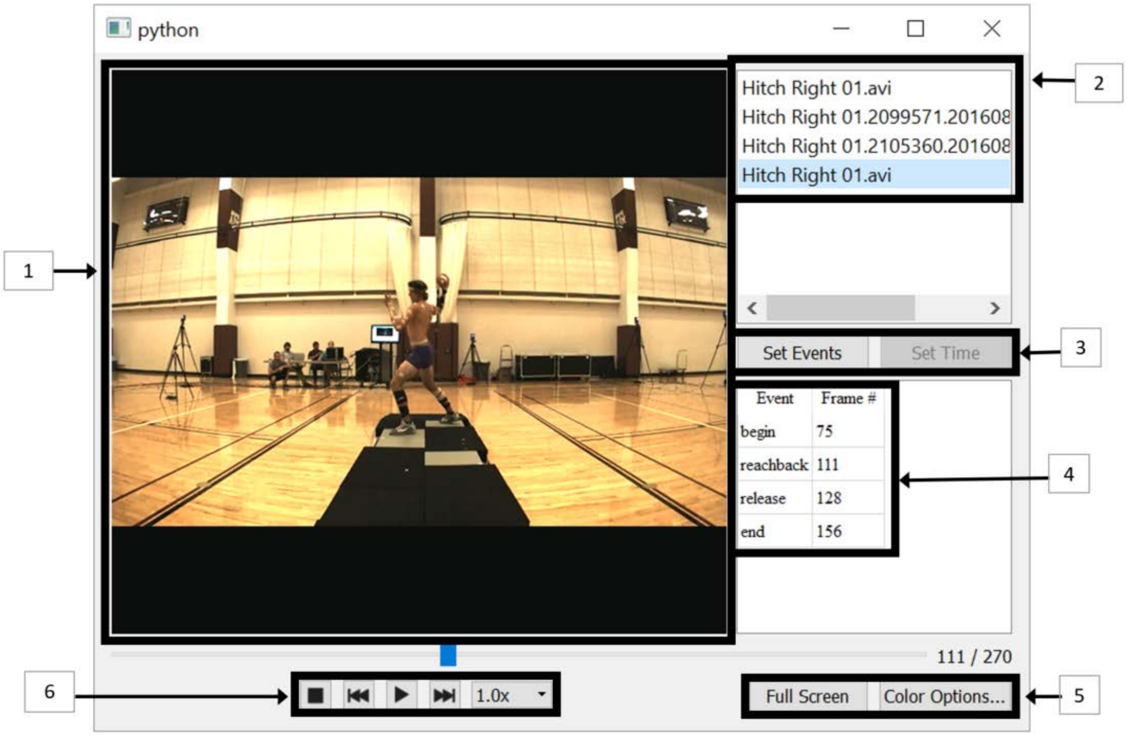


Figure 19: Video player interface with main features highlighted: 1) Main video display, 2) List of available videos for the current trial, 3) Buttons for setting/changing the event times, 4) Events found for the current set of analyses and the frame in which they occur, 5) Options to view full screen or change video color settings, 6) Playback options and a changeable playback rate.

4.3.3 Analysis Modules

The purpose of the analysis modules is to provide an easy to follow framework that can be used for researchers wanting to implement their own custom analysis for a MOCAP application. In order to be compatible, and executable from within the main UI, each analysis script must contain a specific set of functions and attributes including:

- Attributes

- **Results dictionary –** Python dictionary holding any numerical results from running the analysis script. These are the results that will be eventually outputted for further statistical processing.
- **Display dictionary –** Python dictionary containing any results or graphs from the analysis which the user wishes to display in the main UI.
- **Reporting dictionary** – Python dictionary containing all images and descriptions the user wants to be in the final generated PDF report for the analysis. The generic report function requires one image, one figure caption, and a description of the analysis.
- **Description string –** a string describing the analysis that was performed and what is displayed in the final PDF report.
- **List of strings for required events –** a Python list containing individual strings for each event required to perform the analysis

- Functions
    - **Initialization function –** initializes the object and all required attributes of the analysis module. Takes in a filename, and an instance of the dataAnalysis class as parameters.
    - **String function –** returns the name of the module as a string for UI display purposes.
    - **Run function** – contains all functionality required to perform the analysis and returns a dictionary containing the results from running the analysis.
        - Requires an eventTimes parameter (supplied from the main UI).
        - This is the function that is called when the user clicks the 'Analyze' button in the main window.
    - **Generate report function –** a wrapper function that takes in the reporting dictionary and uses it to call the parent classes "generate report" function.
        - This is done as a wrapper to the parent class so each analysis module can use the same 'generate report' function. If a user wants

to edit the way the PDF report is created all they have to do is write it here rather than call the parent classes generate report function.

The programmer can add additional functionality or new attributes as is required for a specific analysis, but in order for the UI to be able execute the analysis script without error each of the listed parameters and attributes must exist. A skeleton framework was created to provide guidelines of how one of these analysis scripts will look, as well as to give some descriptions of the required parameters. This framework is shown in Figure 20. Additional documentation providing more detail on how to implement new analysis scripts will be provided for future users.

```python
"""
@author: Hunter Storaci
Example code for how to create an analysis script for use with the user interface
"""
#If there is a parent class, import it here
from Analyses.FootballDataAnalysis import *
#Class name must match the name of the file
class ExampleClass(FootballDataAnalysis):
    def __init__(self,fname,analysis):
        '''
        This function initializes all of the different parameters that are needed to perform an analysis
        '''
        #Must make this function call to initialize the parent class
        super(ExampleClass, self).__init__(fname,analysis)
        self.dataAnalysis = analysis
        #This next object contains all of the c3d file data from the current trial
        self.c3d = self.dataAnalysis.c3d
        #This next object is an instance of the Trajectory class which is used for manipulating the data
        self.tr = self.dataAnalysis.tr
        #The results dictionary contains the numerical results that will be used for statistical processing
        self.results = {}
        #The display dictionary contains whatever the user wants to be displayed in the user interface
        self.display = {}
        #The report dictionary contains whatever information the user wants to be displayed in the PDF reports
        self.PDFreport = {}
        #self.events contains a list of strings that naming any events needed for the current analysis
        self.events = ['someEvent','someOtherEvent']
        #this is the description of the current analysis that will show up at the top of the PDF report
        self.description = "Description of the current analysis"
    def __str__(self):
        '''
        This function just returns the name of the class as a string
        '''
        return "ExampleClass"
    def run(self,fignum,eventTimes):
        '''
        This function is what is called from the UI and where the programmer can implement a custom analysis
        and return the results back to the main user interface
        '''
        ##########################################################
        #Insert code for analysis here
        ##########################################################
        #This is where you give a plot or other result which will go in the final PDF report
        self.PDFreport['Report Image'] = []
        #This is where you set the caption for the plot that is going to be in the PDF report
        self.PDFreport['Report Caption'] = 'Caption for whatever plot you plan to display'

        #Return both the results dictionary, and the display dictionary
        return self.results, self.display, self.PDFreport

    def report(self,coverFlag,displayDict):
        '''
        This function calls the parent class report generator function
        To create a custom report for the programmer simply needs to override this function with custom code
        '''
        self.dataAnalysis.generateReport(coverFlag,self.__str__(),self.description,displayDict['Report Image'],
                                         displayDict['Report Caption'])
```

Figure 20: Skeleton file of an analysis script that future developers can follow to create custom analysis scripts that are compatible with the software package.

4.3.4 Data Storage

Data storage is performed using Python's object serialization protocol: Pickle [31]. Using this process allows for storage of Python objects in a relatively small size, maintains their current state, and allows access to them again at a later time or from a different location. After a trial has been analyzed, viewing the results is as simple as reloading the pickled object associated with that trial, rather than having to re-run each analysis module every time the results need to be displayed. In the current software platform there are two types of objects that are stored using this method: a project data object and a trial data object. Whenever a C3D file is opened from the UI, the software first checks to see if there are pickled objects stored in memory for the current trial and current project. If those objects exist, the software loads those pickled objects and displays all available information in the analysis window of the UI. This ensures that analyses are not re-run on a C3D file that has previously been analyzed using the software. If no pickled object exists, the UI creates new pickled files for the current project or trial that will be used to store any results that are created for them.

Project Data

The project data object contains information that is specific to a project (i.e., QB throwing biomechanics) but not to the trials that are associated with that project. Information such as the project title, the list of analysis modules associated with the project, images and figure captions that appear on the cover page for reports over the project, the names of the metadata for the project, as well as the functionality that is required for generating the cover page is stored within the project data object. If the project object does not exist, the software instantiates a new object of the project data class and assigns it to the current project. The first time through, the user may be asked to input some basic information about the project needed to generate the PDF reports. The data will be saved in the projects pickled object so that the user will not have to input it again.

Trial Data

The trial data object contains all of the information that is specific to an individual trial rather than to the project that the trial is associated with. This is information such as the C3D filename, the list of analyses that have been executed, values for the metadata, numerical results from running the analyses, results to be displayed in the UI, or the list of events and their times. When the user selects a C3D file of a trial they wish to open, the software searches to see if a pickled object for that trial exists. If a pickled object is found and it contains results, they are automatically displayed in the UI. From that point the user can run additional analyses, re-run analyses, edit metadata, etc., and the trial object will be updated and saved when any changes occur. If a pickled object does not exist, an instance of the trial data class is created for the current trial and the UI is updated to show that no analyses have been run for that trial.

4.3.5 Software Plugins

Future researchers may want to extend the functionality of the software package to add additional features. Potential ideas for these extensions include: a video editing GUI, a framework to assist researchers in performing statistical analysis, extensions to the PDF generation functionality, or any number of things that could be of use to researchers in the future. To allow for extensions to be added to the framework without requiring future users to edit the original source code, a method was created for extending the package through the use of software plugins. Plugins can be created in an environment that is completely separate from the original framework, and then added to the 'Plugins' subfolder within the source code architecture. Each time the main UI is loaded, plugins are dynamically loaded and built from this subfolder, and a command for accessing the plugin is added to the 'Tools' section of the main UI toolbar. Every plugin receives two parameters from the main UI upon their instantiation: a trial object, and an instance of the main UI. These two parameters will allow the plugin to access all of the necessary features and attributes of the main UI, as well as all of the classes that are imported by the main UI, in order to program the necessary features of the plugin. This will allow future programmers to create their own plugins, and upon adding it to the correct subfolder see the plugin available for use from the main UI. As an example of this process, two plugins were built and added to the software framework: a batch processing plugin, and a plugin for compiling and outputting final results.

Batch Processing

The batch processing plugin allows the user to load and analyze multiple c3d files at one time, reducing the time required to analyze multiple trials for a MOCAP application. This can be a significant time reduction; for certain studies large quantities of c3d files need to be analyzed. For example, each participant in the QB throwing biomechanics study has anywhere from 50 to 80 trials to be analyzed. An image of this plugin is shown in Figure 21.

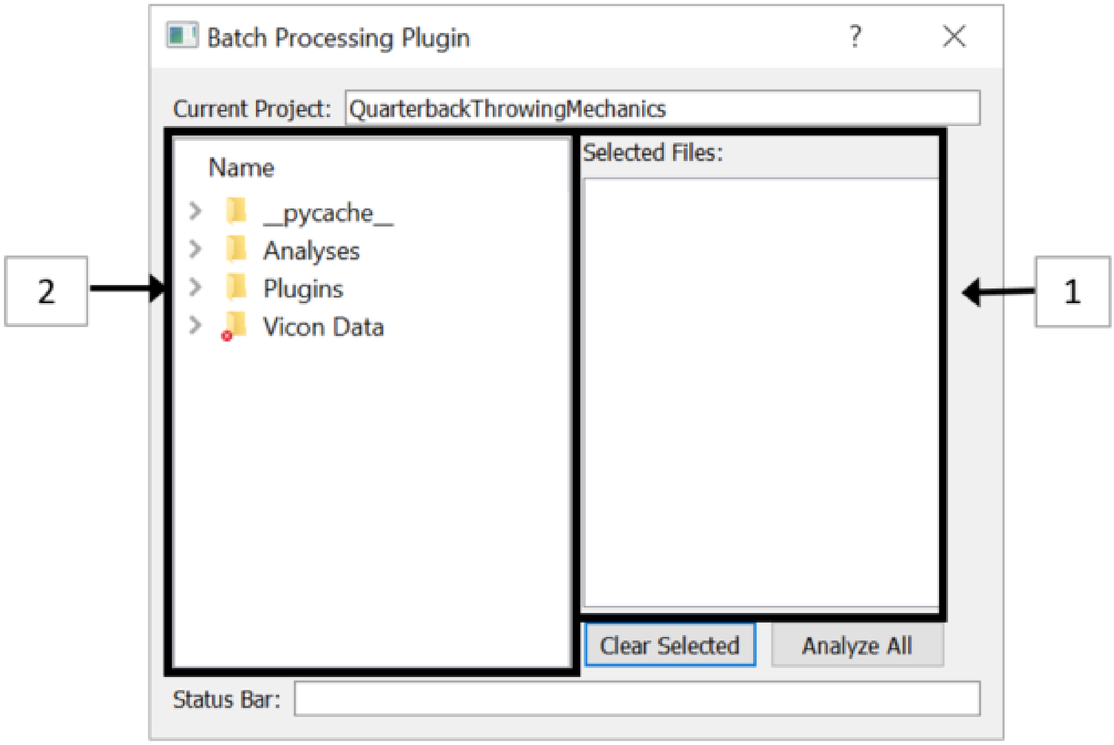


Figure 21: Batch processor plugin screen capture with features highlighted: 1) Display where all currently selected files will be shown, 2) File tree where the user can select multiple trials to be analyzed.

Results Output Plugin

The results output plugin allows users to select a subset of pickled trial objects, which have already been analyzed using the main UI, and compile and save the results into an output file which can be opened elsewhere for additional statistical processing of the data. The final output file is a Pandas DataFrame [32] which will allow for use of the Pandas Python module for users when performing any downstream processing of the results. This final results output contains no dependencies on the source code for the software platform.

Therefore, the results can be opened for processing on any computer regardless of whether or not it has access to the main UI and the source code that was used to create the results. A screenshot of the results plugin is shown in Figure 22, and an example of what these results could look like is shown in Table 3.

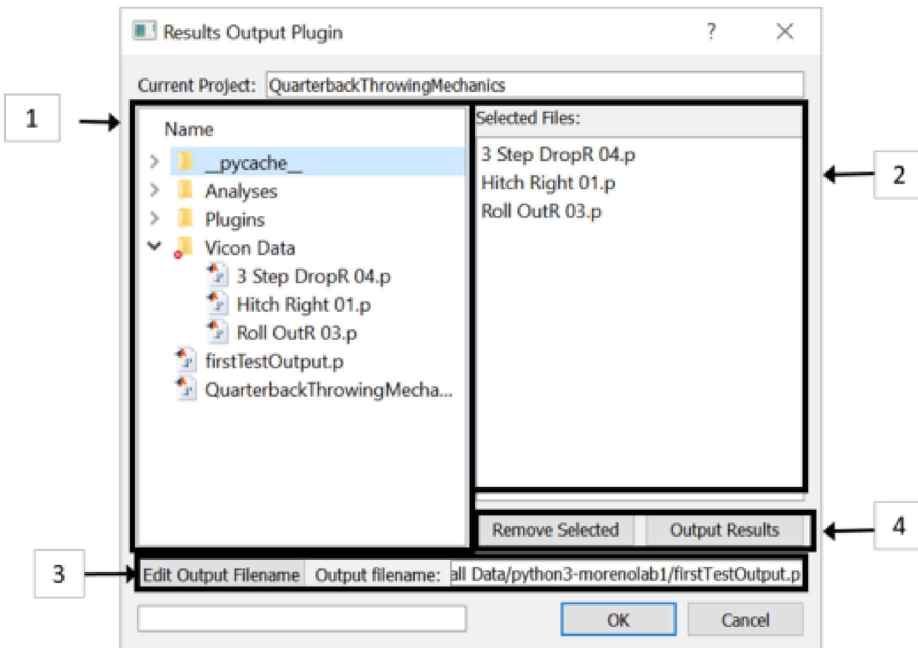


Figure 22: Screenshot of the results output plugin with the main features highlighted: 1) File tree where the user can select the results files that they wish to compile into an output, 2) Display of the selected results, 3) Button for the user to set a filename and a save location for the results, 4) Buttons to remove a file from the selected files, or to compile the results and save to the specified output file.

Index	School	Age	Weight	Velocity (MPH)	Release time (s)	Etc.
3-step drop	College Station	13	145	33	0.5	…
Hitch	Bryan	16	170	41	0.44	...
Rollout	Rudder	18	215	47	0.39	…

Table 3: Example of what a DataFrame output could look like, and the type of information that it could contain. Each row contains all of the outputs for a given trial. The columns contain all results and metadata that were included with the output. A Pandas DataFrame can include as many rows and columns as the user needs in order to output all of the relevant information for a trial.

4.3.6 C3D Data Interface

The c3d parsing class takes in the filename of the current c3d file, parses the file, and saves all of the relevant information to variables that can then be accessed within the user-programmed analysis modules. The stored information includes: the number of frames in the trial, the frame rate of the cameras and force plates, the trajectory data and error information, a list of all of the marker names, all analog data information, information about the coordinate system of the capture volume, information about the force plate locations within the capture volume, and the information about the numbering of the force plates. The information stored in the c3d file is all the information about a motion capture trial a user needs to program custom analyses.

4.3.7 Trajectory Datatype

The trajectory datatype class is written to contain the necessary math functionality for a user to program custom analysis modules for a given motion capture project. A trajectory datatype object has two parameters: data and errors. Data contains the 3D point data corresponding to a markers coordinates in 3D space, and errors contains the residual error information provided by the motion capture system that corresponds to the relative accuracy of the 3D coordinate measurements. The class contains wrappers to NumPy math operations and allows the user to access all of these operations while also maintaining correct error propagation of the residuals. For example, if a user wanted to find the average position of two trajectories, right hip and left hip, they would write: $hip\ average = (right\ hip + left\ hip)/2$. The result of this operation, hip average, would contain a trajectory datatype object that contains a data attribute, which is the average position of the 3D point data, and an error attribute, which is the new residual error information after performing mathematical manipulations of the data. In addition to basic math operations (i.e., add, subtract, multiply, divide, average), the class also contains some more advanced trigonometric functionality and functionality the user can use to calculate angles between different markers. The user can access all of these operations the exact same way they would access a NumPy math operation, but when they call these operations on a trajectory datatype they get both the result of the operation and the relative error after performing that operation. This is a useful too because oftentimes creating custom analysis of data requires a user to perform a lot of different mathematical manipulations of the data. Having some idea of how the error was propagated throughout all of these calculations will give researchers a good idea of how confident they can be in their results.

4.4 Discussion

The goal of this software package was to create an interface and a set of tools to assist future researchers in implementing their own custom analysis of MOCAP data. Following this model, the software needed to as generic as possible so as to not limit the scope of applications to which it can be applied. This was done by establishing a framework for creating custom analysis scripts of MOCAP data, and creating a user interface which researchers can use to access data, view video files and results, and run their custom analysis scripts to analyze project-specific data. As long as future developers follow the required class syntax and save their file in the correct directory, there is no limit to what analyses can be performed. The UI provides the means for viewing the results and generating reports over the results and can either be used as a tool for presenting the data to collaborators or in the development of new custom analyses. As more and more researchers cycle through the lab and work on different projects we will begin to build up a database of Python analysis scripts that new researchers can use as a basis to build off of for the creation of new analyses which they wish to perform. Because all of the files will follow the established framework, it will be easy to understand what was done and to execute old files and view the results using the UI, even after the original programmer has moved on.

As with any software platform, there will more than likely be need to expand the functionality and add features to the software in the future. To this end, a framework has been provided for creating plugins to the main UI. These plugins are dynamically loaded when the user loads the main window, so additional plugins can be added at any time. Two example plugins, one for batch processing of data files, and one for compiling and outputting results, were created to provide an example of how to implement future plugins for extending the software.

To not limit the operating system (OS) requirements for the software, a cross-platform GUI framework was used and the software will be packaged as an executable file that can be run from any OS. This leaves the door open to the possibility of distributing the software to potential users or collaborators or creating a limited version of the software that could be given to collaborators along with polished results and videos a tool for them to be able to view the results and video files.

CHAPTER V

SOFTWARE VERIFICATION AND VALIDATION

5.1 Software Package

To validate the software package discussed in Chapter 4, a sample analysis for of the three types of throwing trials performed in the QB Mechanics study was performed and an example PDF report was generated over the analysis. The software was able to handle all three different types of throwing trials: a hitch, a 3-step drop, and a rollout that did not contain any analog data. Each of these trials was then successfully analyzed simultaneously using the batch processing plugin, and the results were compiled and saved using the results output plugin. Further testing of the software package was performed by successfully implementing, displaying, and saving results for an analysis that was written by undergraduate students in my research lab to perform analysis of data for a different MOCAP study.

5.2 Quarterback Analysis

The validation of the software using data from the QB study included performing analysis for some of the coaching points discussed in Chapter 3 (as well as a couple of additional analyses deemed useful by the researchers), and using graphical displays to visualize the results. Some of the coaching points were obvious on how to display graphically, while others were not as intuitive and took a little bit of interpretation of the coaches intentions to visualize. The analyses shown are over three different throwing trials for the same subject.

5.2.1 Hand Path/Release Time

Analysis of the hand path and general posture of the quarterback was not explicitly stated in the interviews as a coaching point, but several coaches mentioned looking at specific release characteristics or wanted the players to be standing erect at the moment of release. The following plot was created to give a visual depiction of what the player's stance looked like at key instances during the throwing motion. There were no quantitative results of the players' posture returned from this analysis, however this script was also used to compute the release time for the throw, which was a commonly mentioned coaching point from the interviews. Results from this analysis are shown in Figure 23, Figure 24, and Figure 25.

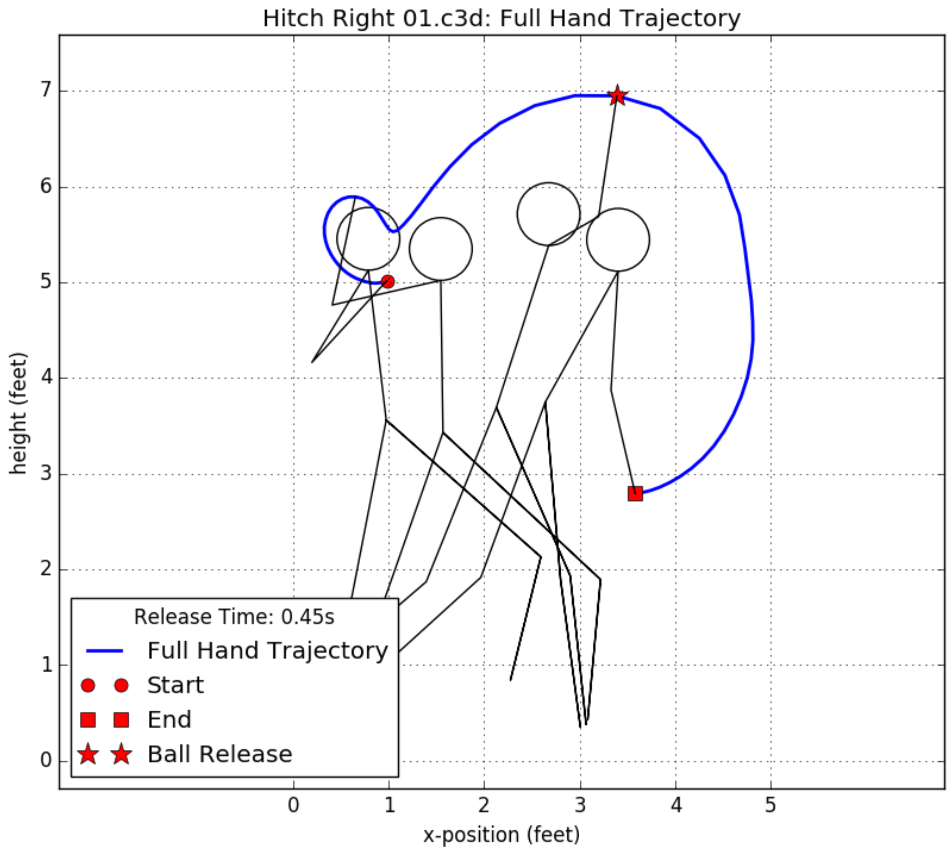


Figure 23: Display of the forward hand path, and general body posture for a subject throwing a hitch pattern off of no drop back to the right side of the capture volume.

Figure 24: Display of the forward hand path, and general body posture for a subject throwing a corner pattern off of a 3 step drop back to the right side of the capture volume.

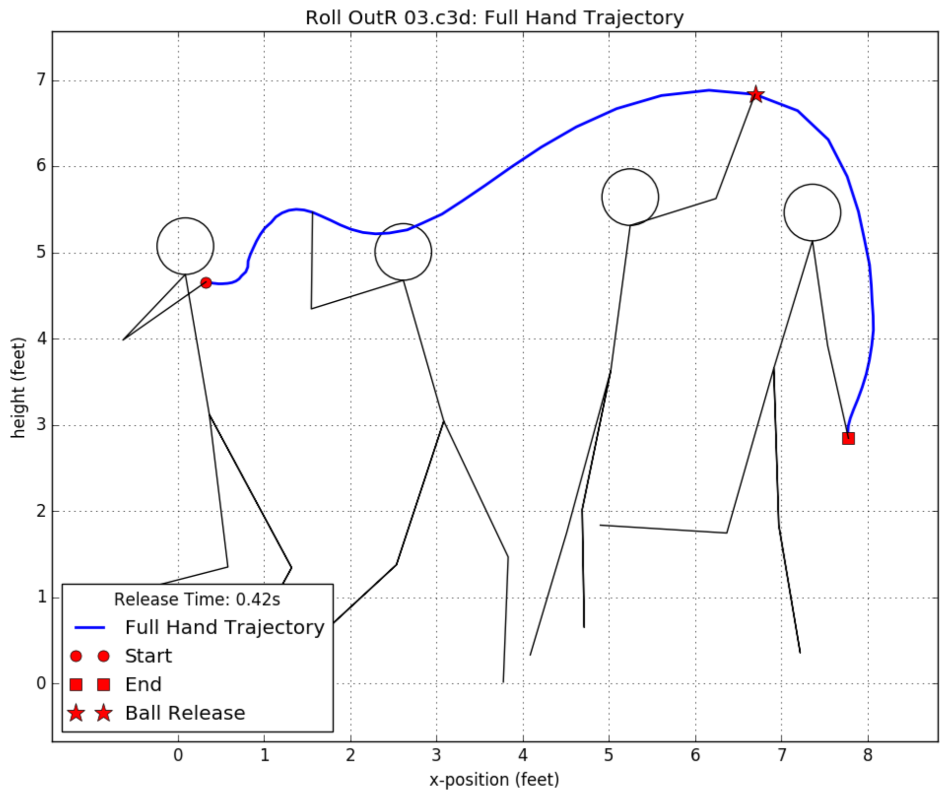


Figure 25: Display of the forward hand path, and general body posture for a subject throwing comeback pattern off of a rollout to the right side of the capture volume.

5.2.2 Hip Leading Angle

The hip leading angle was one of the most common coaching points mentioned during the interviews with coaches. What they are looking for is for the hips to “lead” the shoulders throughout the first part of the throwing motion and for the shoulders to catch the hips just at the moment of ball release. This means they are looking for a hip leading angle of zero degrees at the moment of release which says the shoulders are perfectly square with the hips. Although not explicitly stated the hip leading angle is representative of a well-known phenomenon that occurs in the throwing motion: the kinetic chain. The kinetic chain is the synchronous use of selective muscle groups to transfer energy from the lower body to the upper extremity [34]. Coaches may focus on the hips leading the shoulders as it is one of the easier pieces of this chain to visually inspect and coach. Results for this analysis are shown in Figure 26.

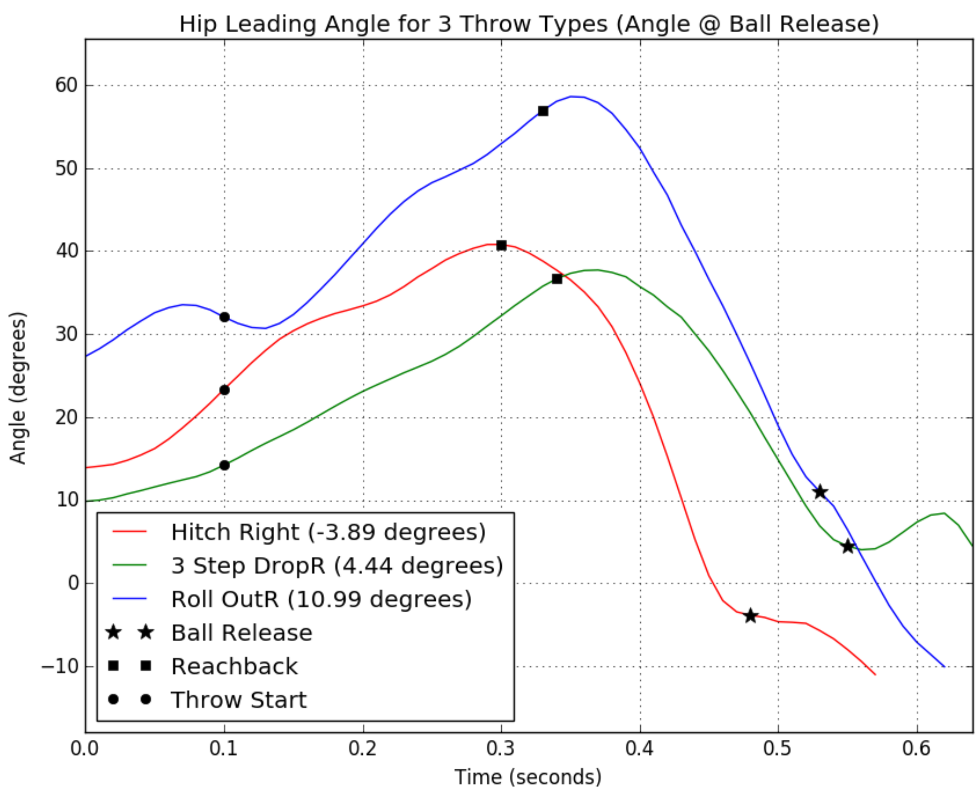


Figure 26: Plot of the hip leading angle for a single subject performing each of the three types of throws in the study.

5.2.3 Speed Chain

Another method of visually displaying the kinetic chain is by comparing the angular velocities of the hips, shoulders, and hand. While not explicitly mentioned in the interviews, this plot represents some key principles that are thought to represent good throwing form. While the hip leading angle plot shows the relative angle of the hips and the shoulders, this plot is a way to visually see when each of the different portions of the body reached their peak angular velocity. Ideally, the hips will reach their peak velocity first, followed by the shoulders, then the hand. Results for this analysis are shown in Figure 27, Figure 28, and Figure 29.

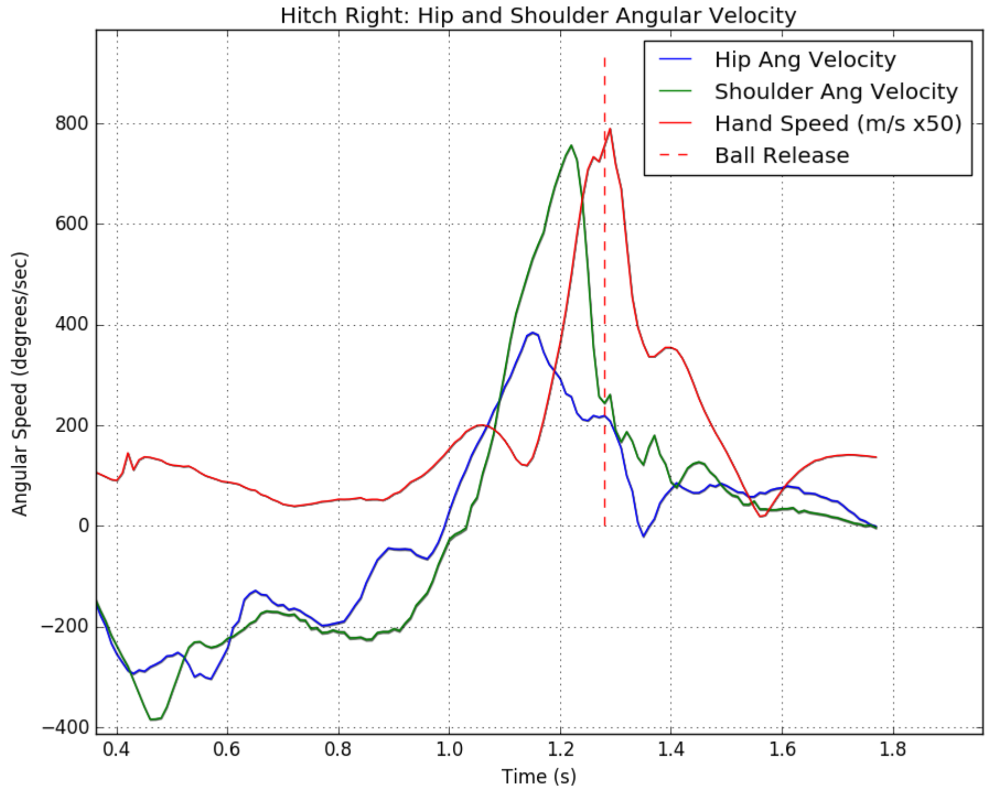


Figure 27: Display of angular velocity for the hip, shoulder, and hand for a subject throwing a hitch route off of no drop back to the right side of the field.

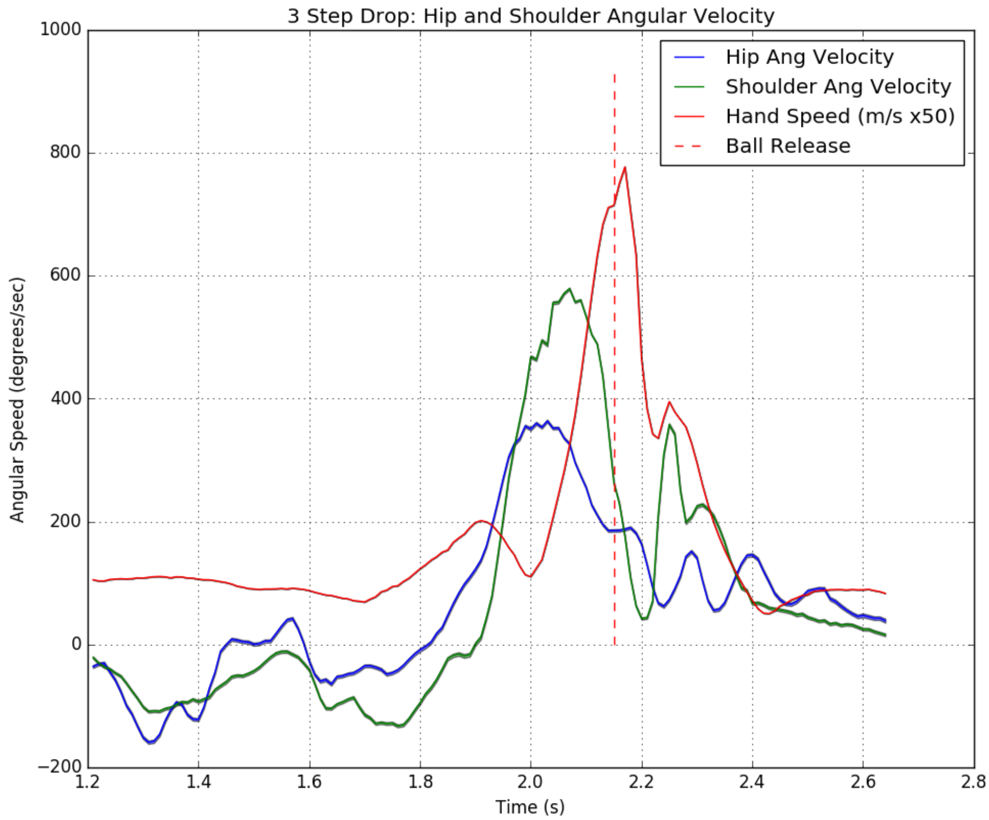


Figure 28: Display of angular velocity for the hip, shoulder, and hand for a subject throwing a corner route off of a 3-step drop to the right side of the field.

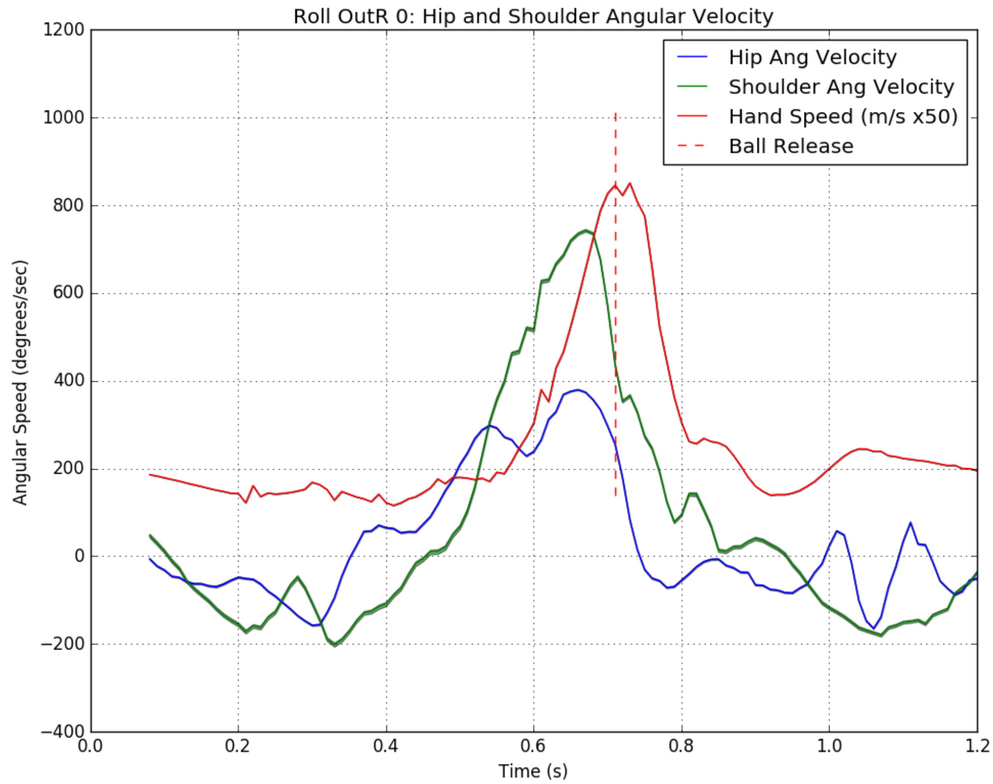


Figure 29: Display of angular velocity for the hip, shoulder, and hand for a subject throwing a comeback route off of a rollout to the right side of the field.

5.2.4 Elbow Flexion Angle

One of the points that was common among the coaches interviewed was the extension of the elbow at the point of ball release. Most of the coaches like to see the elbow nearly fully extended so that the QB is releasing the ball from a high point and is less likely to have the pass deflected or batted down. From a biomechanics standpoint, this means looking at the angle of elbow flexion at the instant of ball release. The measured elbow flexion angle throughout the throwing motion is shown in the figure, but the numerical value at the point of release is what will be used for statistical comparisons between QBs and to evaluate the consistency of a QB. A visual display of this angle is shown in Figure 30 and the results from this analysis are shown in Figure 31.

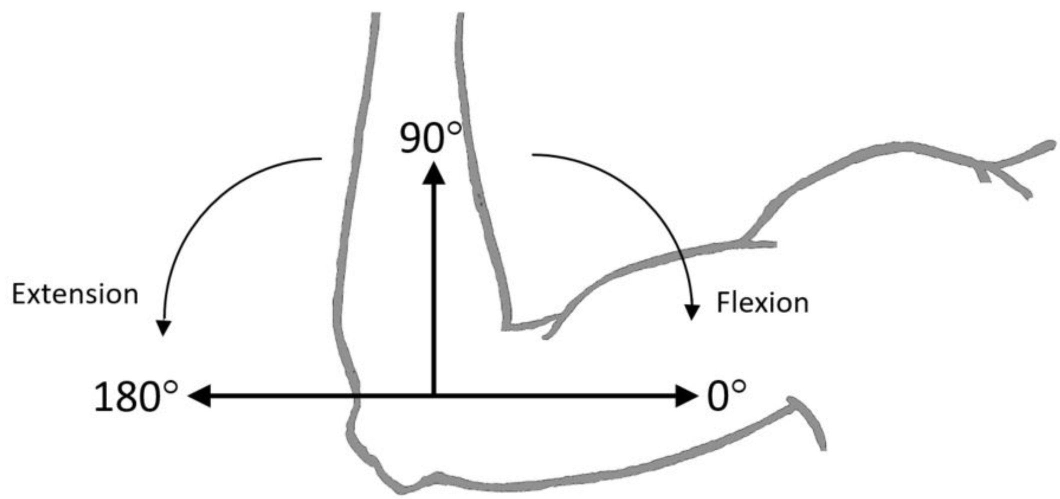


Figure 30: Display of the measured elbow flexion angle.

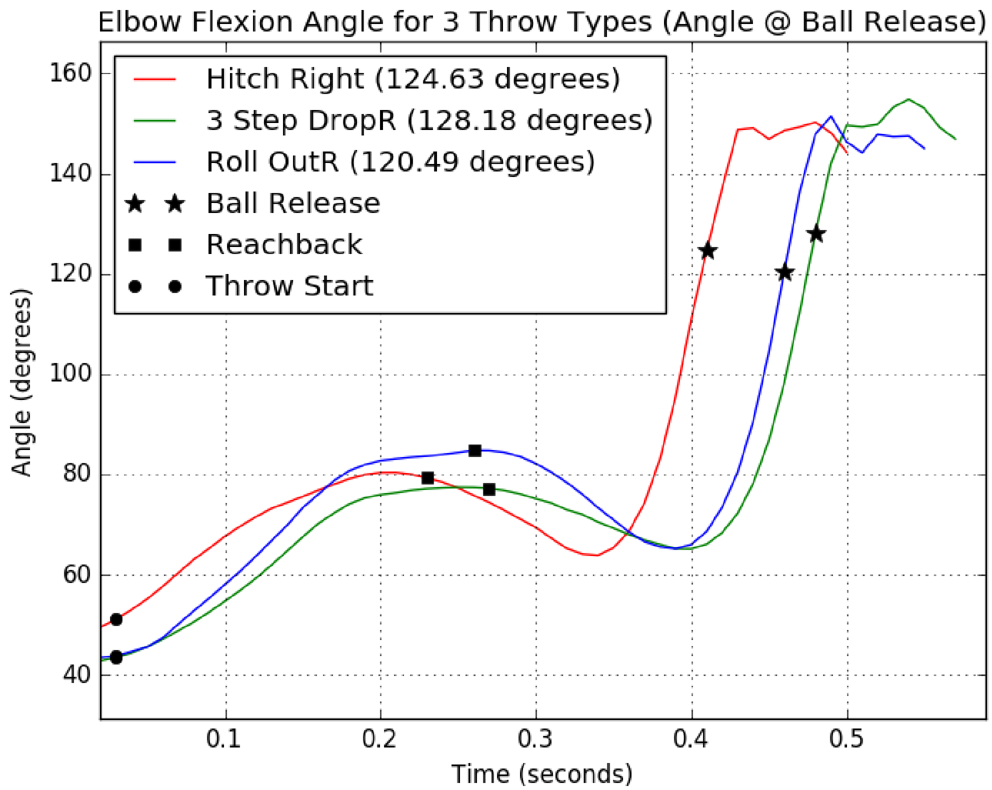


Figure 31: Plot of the elbow flexion angle for a single subject performing each of the three types of throws in the study.

5.2.5 Shoulder Rotation

The rotation of the shoulder was not a point explicitly mentioned by the coaches; however, this element of the throwing motion is another piece of the kinetic chain, which was an underlying theme from the interviews. Additionally, this was one of the kinematic parameters looked at in both of the previous studies of QB throwing mechanics, so performing this analysis for the sake of comparing results was useful. A display of this angle is shown in Figure 32 and the results from this analysis are shown in Figure 33.

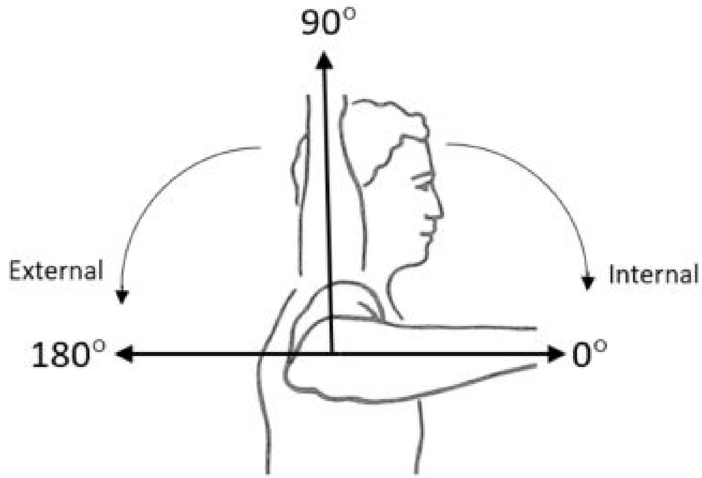


Figure 32: Display of the measured shoulder rotation angle.

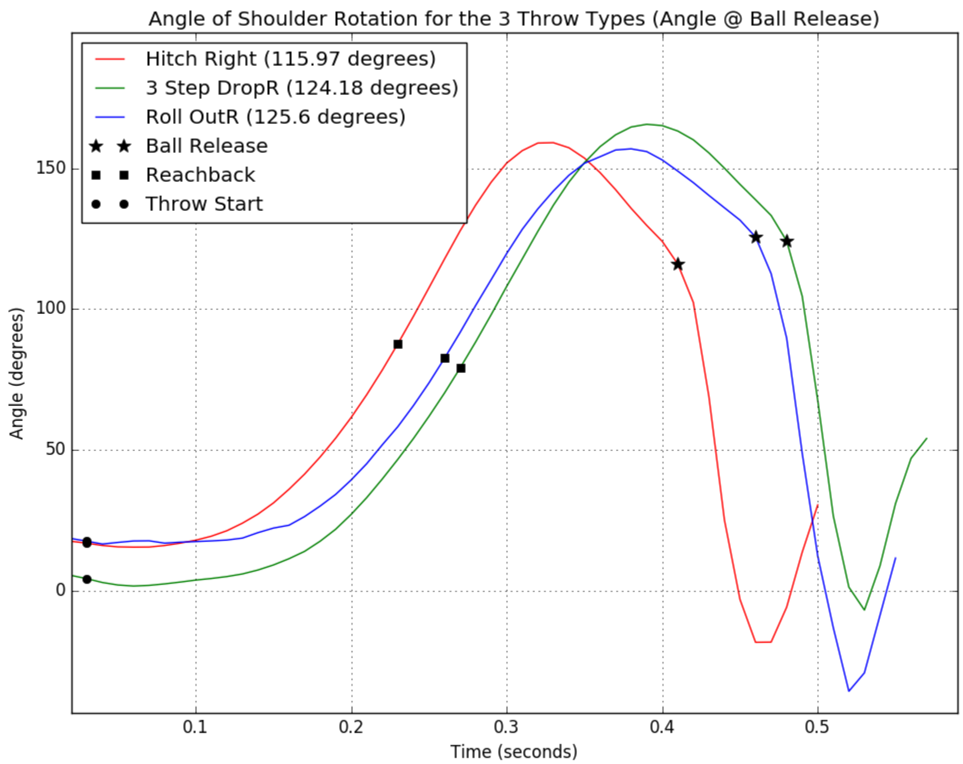


Figure 33: Plot of the shoulder rotation angle for a single subject performing each of the 3 types of throws in the study.

5.2.6 Orientation at Ball Release

There were many different variations and methods among the different coaches for looking at a QB's orientation at the moment of ball release. The common goal among all coaches, however, was ensuring that the QB's hips and shoulders or shoulders and belly button are square to their target at the moment of ball release. According to coaches, being square to the target is one of the leading factors in the accuracy of a pass. The results from this analysis are shown in Figure 34, Figure 35, and Figure 36.

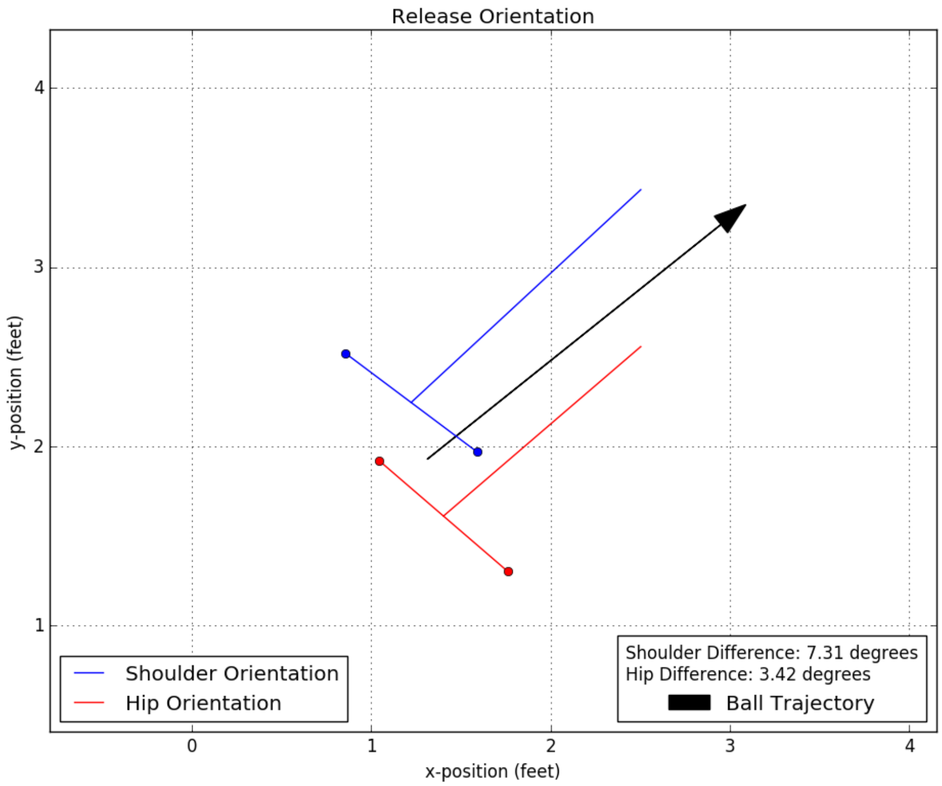


Figure 34: Plot displaying the orientation of the hips and shoulders compared to the ball trajectory at the moment of ball release for a subject throwing a hitch route off of no drop back to the right side of the field.

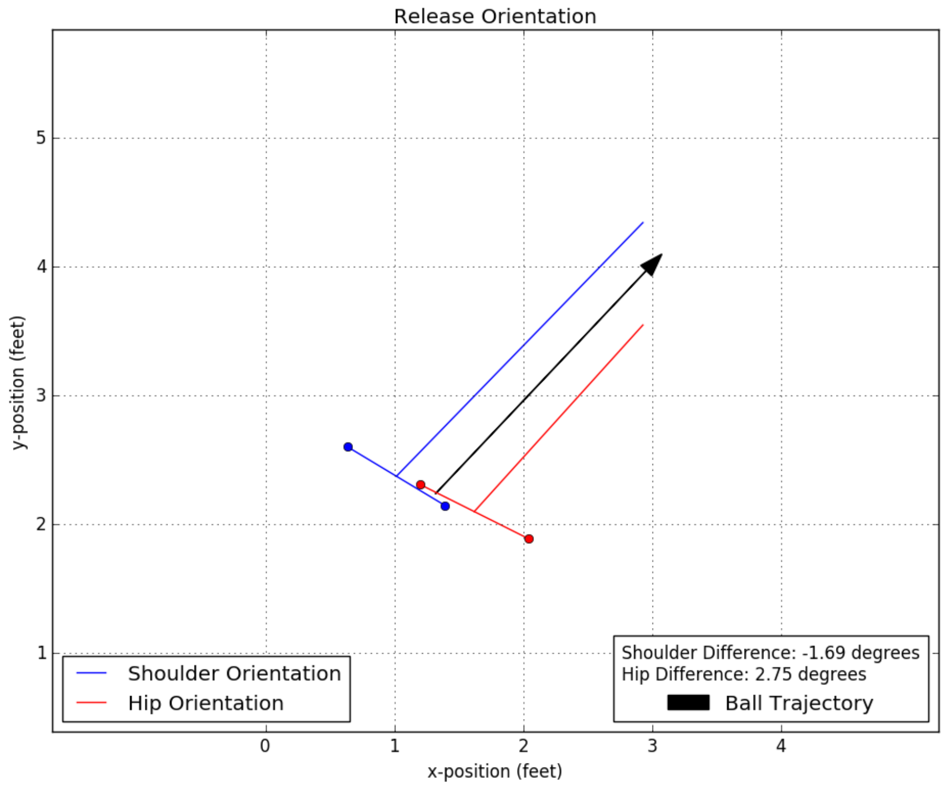


Figure 35: Plot displaying the orientation of the hips and shoulders compared to the ball trajectory at the moment of ball release for a subject throwing a corner route off of a 3-step drop to the right side of the field.

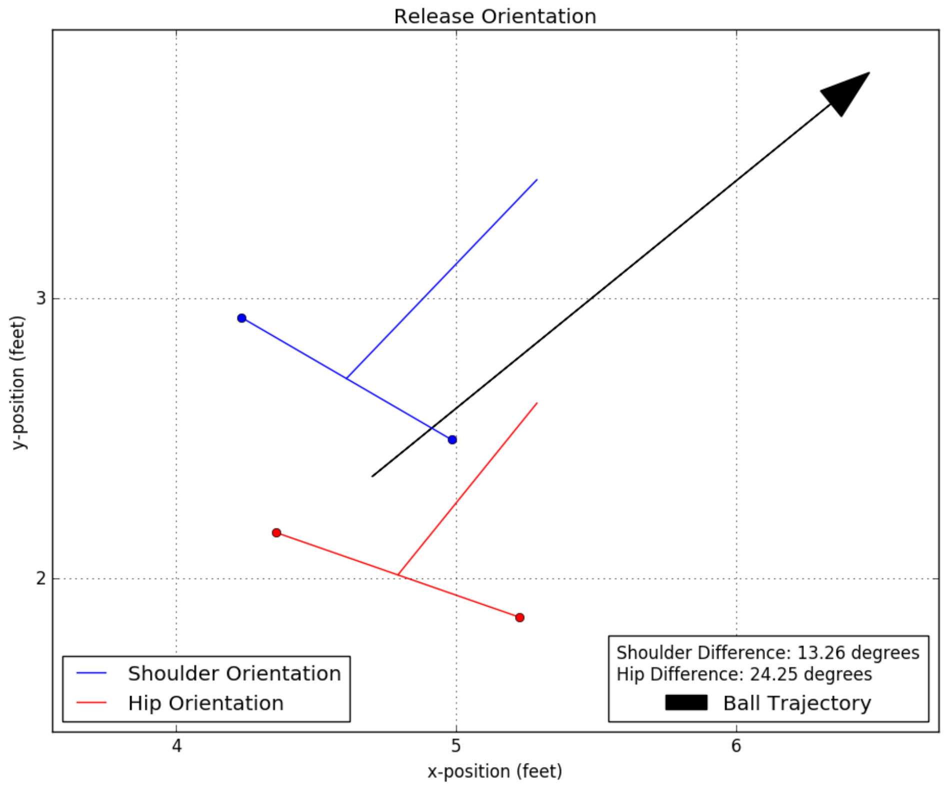


Figure 36: Plot displaying the orientation of the hips and shoulders compared to the ball trajectory at the moment of ball release for a subject throwing a comeback route off of a rollout to the right side of the field.

5.2.7 Stride Analysis

There were a few different variations as to what coaches look for in the stride of a quarterback. Some coaches like for the lead foot to be aimed just to the left of the target, some like the QB to point the lead toe towards the target, some do not ever want the QB to stride out much past the width of their shoulders, and some are more concerned with where the back foot is pointing than the front foot. While there are many different schools of thought on what constitutes "correct", the majority of the coaching techniques can be evaluated by watching the paths of each foot, measuring the length of the stride, and comparing the trajectory of the stride vs the trajectory of the ball. Results from this analysis are shown in Figure 37, Figure 38, and Figure 39.

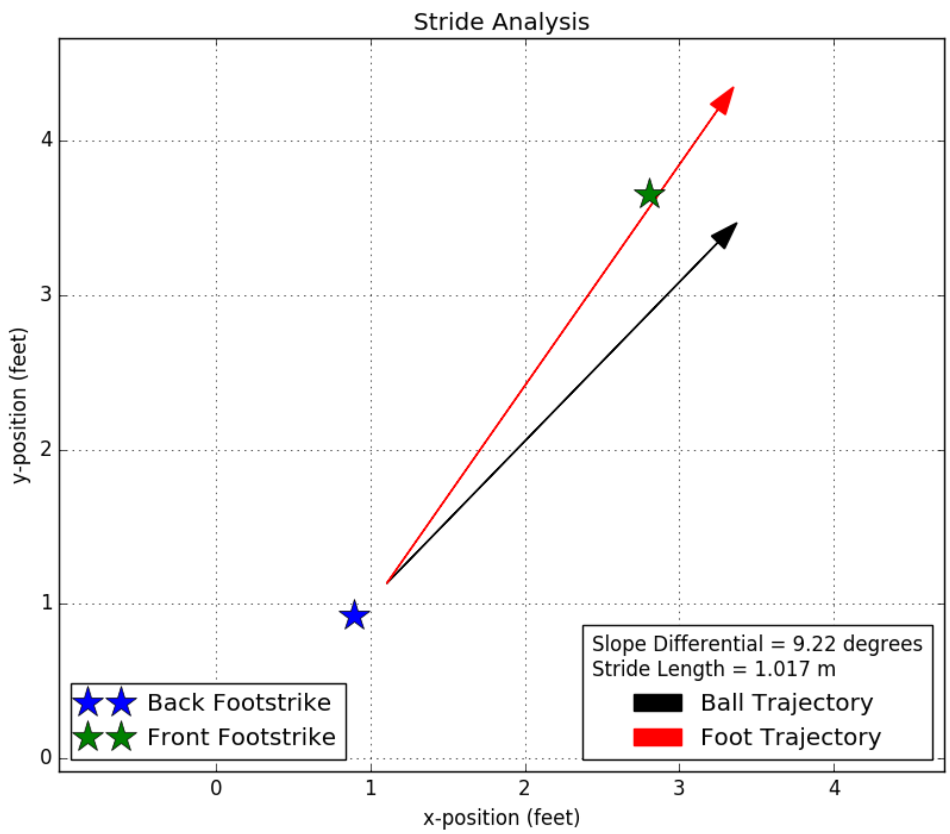


Figure 37: Plot of the foot and ball trajectories for a subject throwing a hitch pattern off of no drop back to the right side of the capture volume.

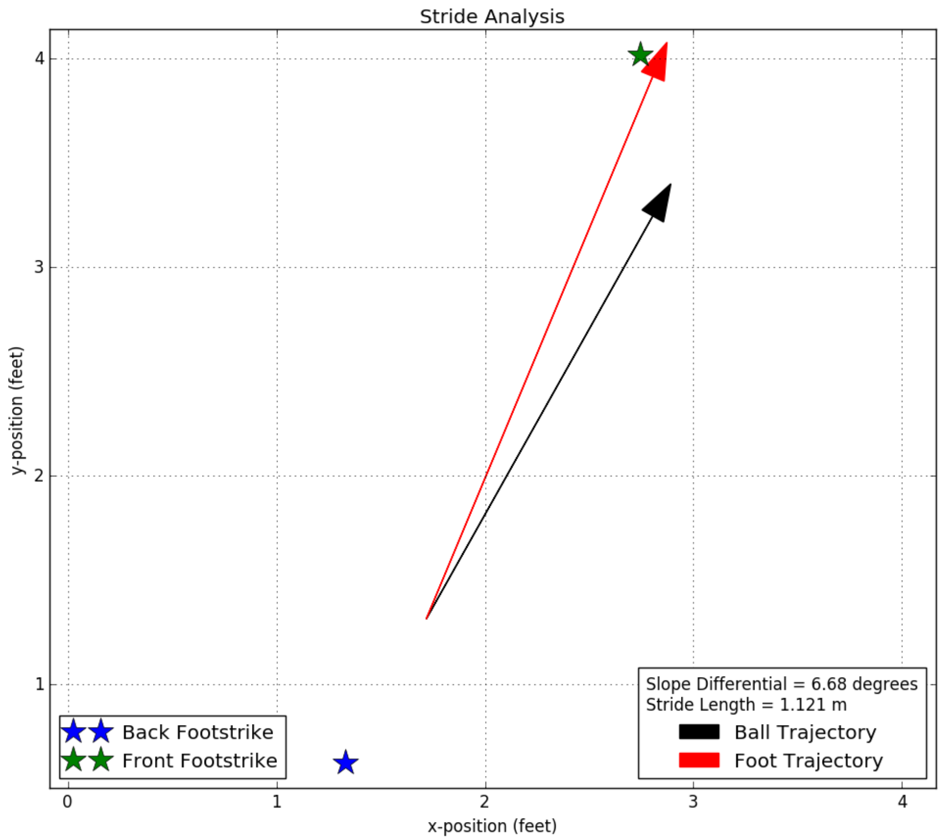


Figure 38: Plot of the foot and ball trajectories for a subject throwing a corner pattern off of a 3-step drop to the right side of the capture volume.

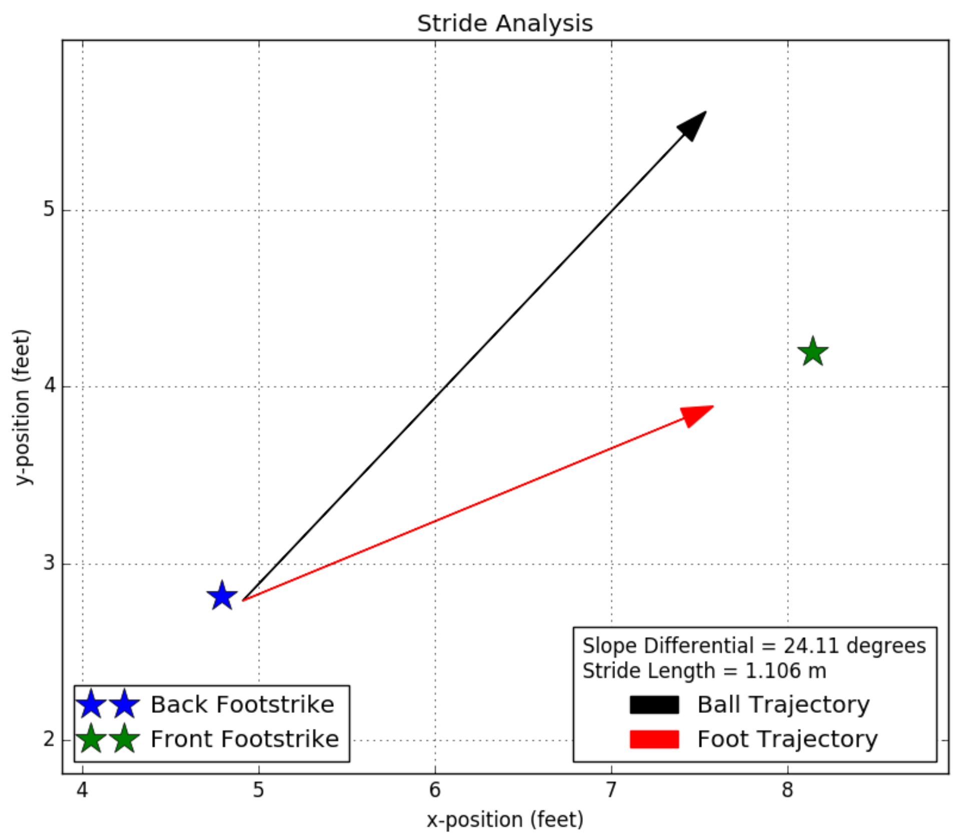


Figure 39: Plot of the foot and ball trajectories for a subject throwing a comeback pattern off of a rollout to the right side of the capture volume.

5.2.8 Weight Distribution

This was another parameter not explicitly mentioned during the interviews, yet several coaches talked about a QB maintaining their balance throughout the throwing motion. Because of this we decided that looking at the weight distribution on each of the QB's feet during the throwing motion would be a useful analysis. Note: no graph is shown for the rollout throwing pattern because that throw did not occur on any force plates. Results from this analysis are shown in Figure 40, and Figure 41.

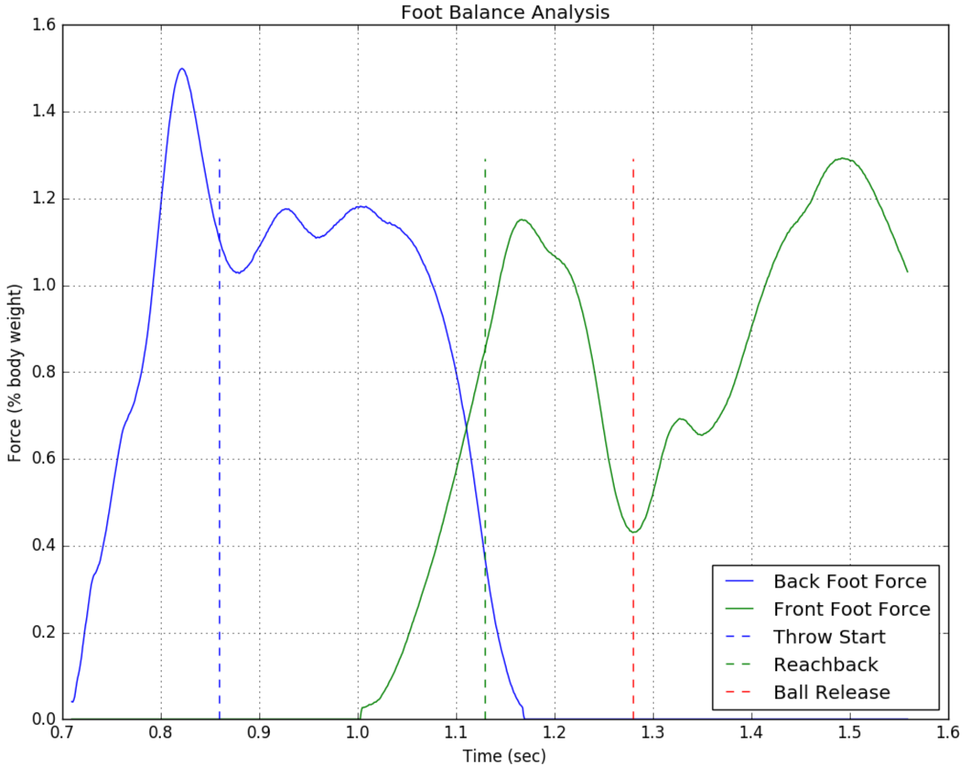


Figure 40: Plot showing the weight that the subject placed on the front and back feet throughout the throwing motion for a subject throwing a hitch to the right side of the field.

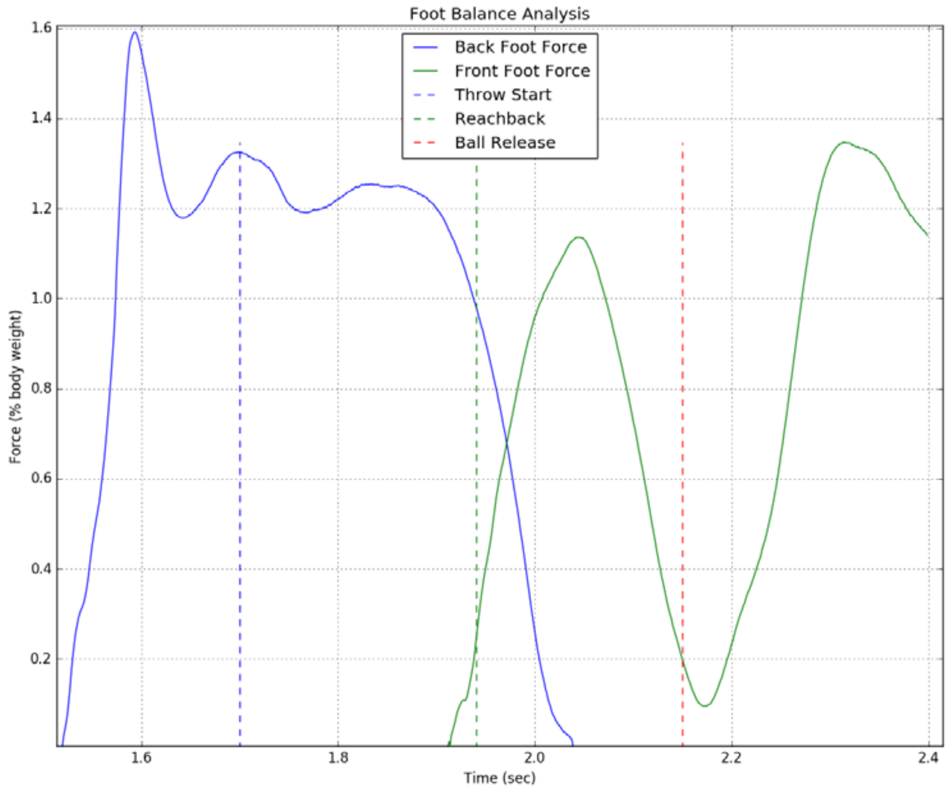


Figure 41: Plot showing the weight that the subject placed on the front and back feet throughout the throwing motion for a while throwing a corner pattern to the right side of the field.

5.3 Additional Analysis

Undergraduate researchers assisting with this project wrote code to perform an additional analysis to analyze the foot strikes of sheep as part of a medical device study in conjunction with Houston Methodist Research Hospital. They first wrote the code as a standalone Python script, and then reformatted it for use with the software platform. The assignment had four different steps:

- Determine the foot strikes for the back two legs of a sheep during a gait trial (Figure 42)
- Determine which foot strikes occurred on the force plates embedded in the walkway (Figure 43)
- Determine if the back foot was the only foot located on the force plate for the duration of the foot strike (Figure 44)
- If all of the above criteria were met calculate the maximum force exerted by the back leg during the foot strike (Figure 45, Figure 46)

Here are the results from their analysis:

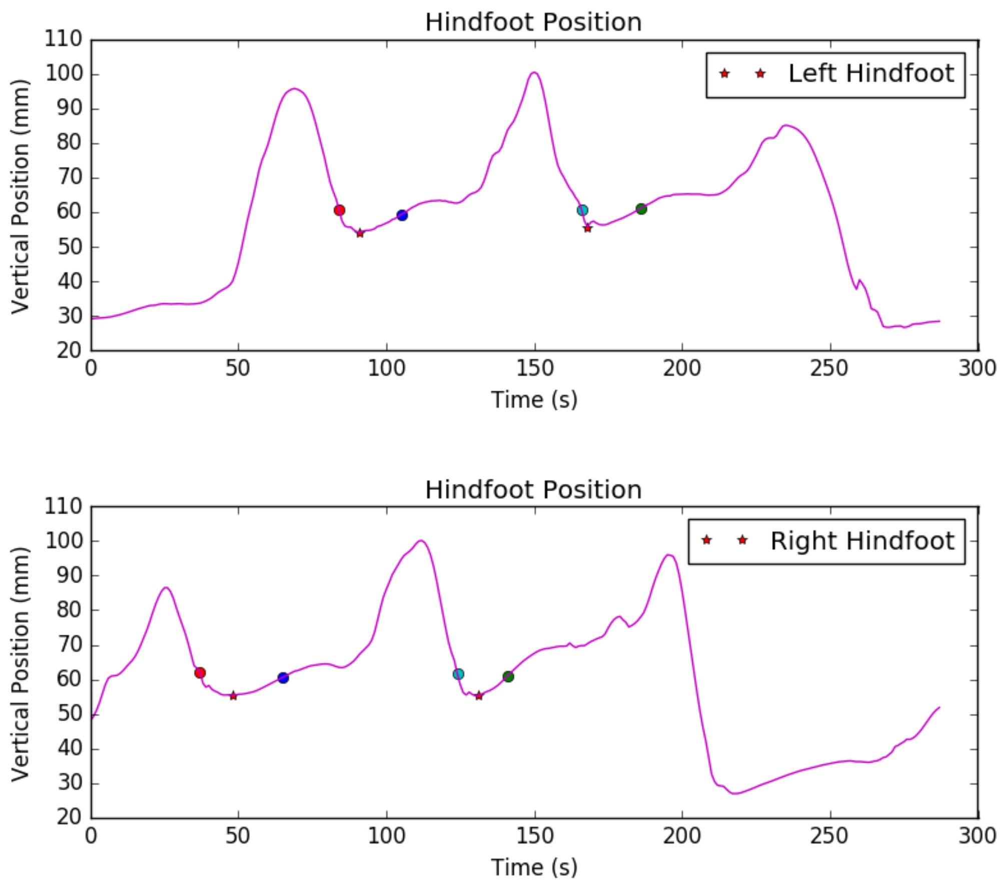


Figure 42: Calculation of the foot strikes for the left and right hind feet of a sheep.

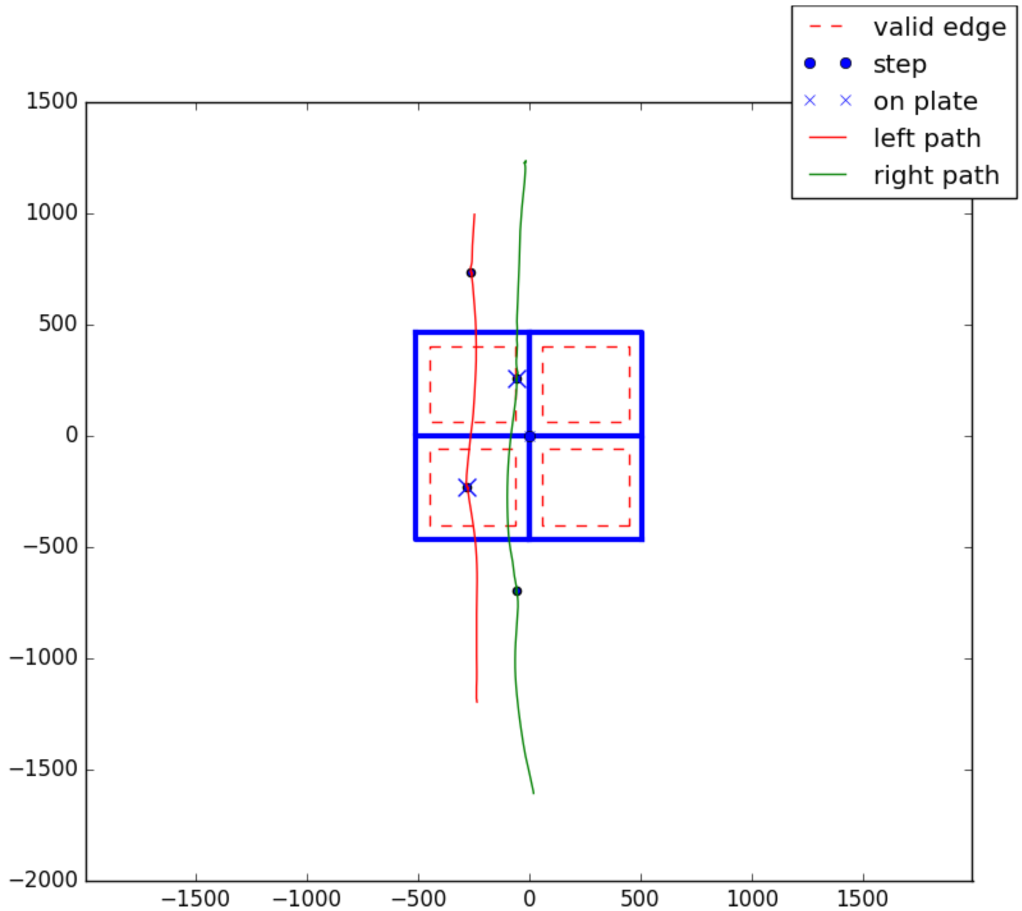


Figure 43: Determining whether or not each foot strike occurred on the force plates and which force plate they occurred on.

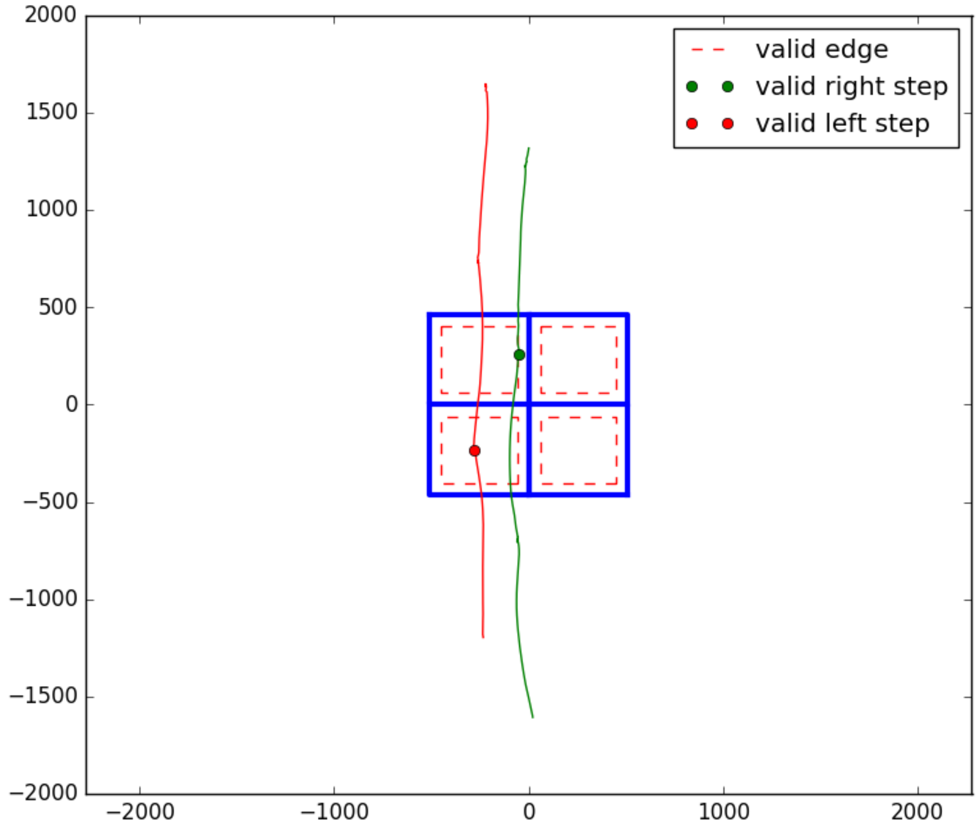


Figure 44: Determining whether each foot was the only foot present on the force plate during the foot strike.

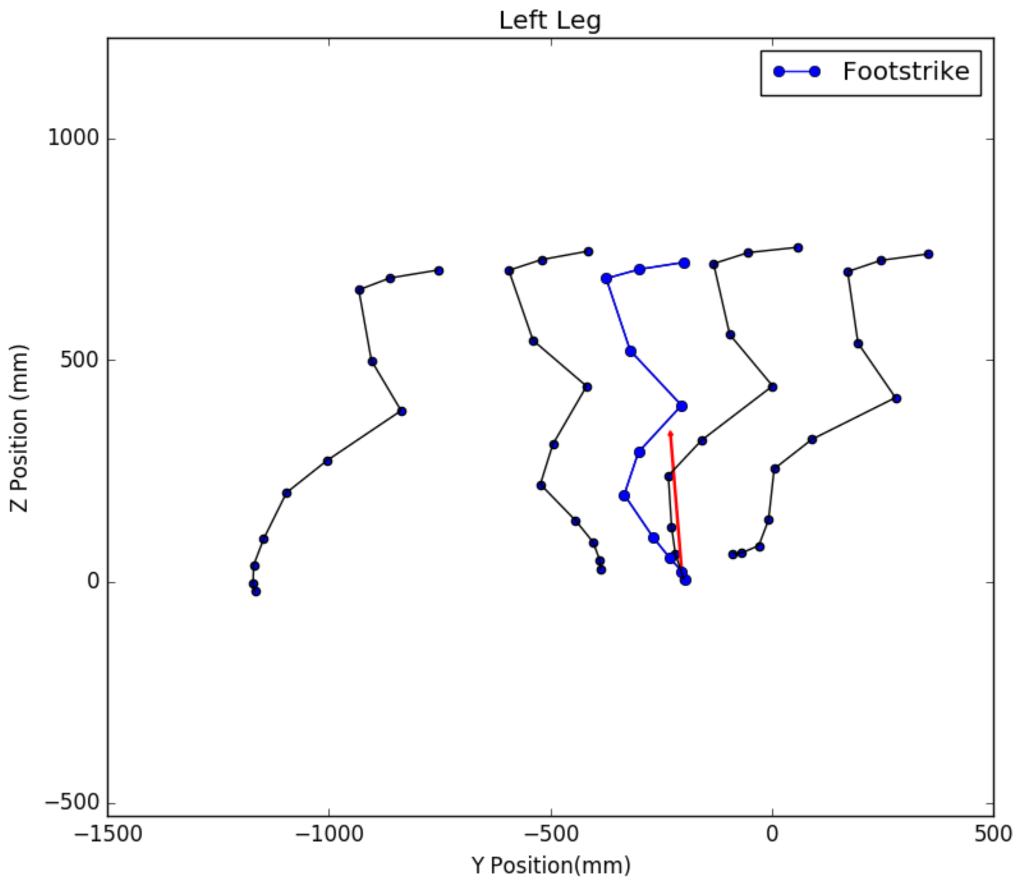


Figure 45: Plot of the swing phase for the left foot of a sheep including the maximum force vector calculated during the foot strike.

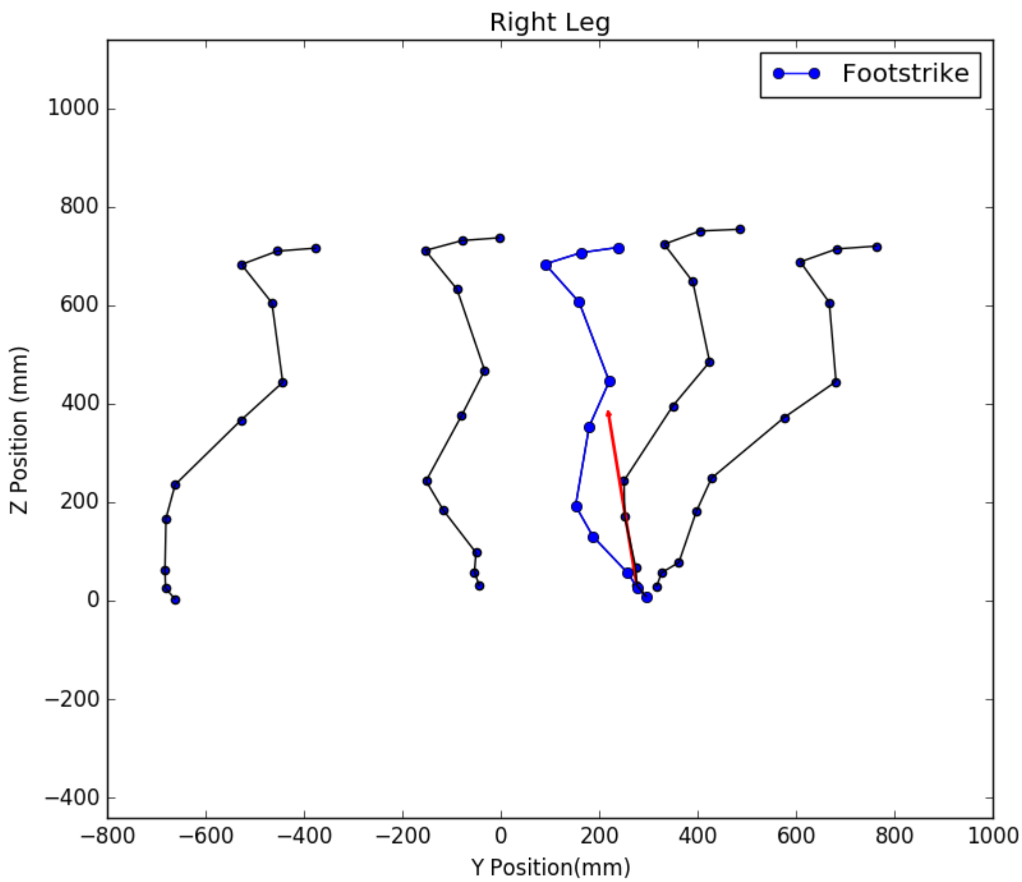
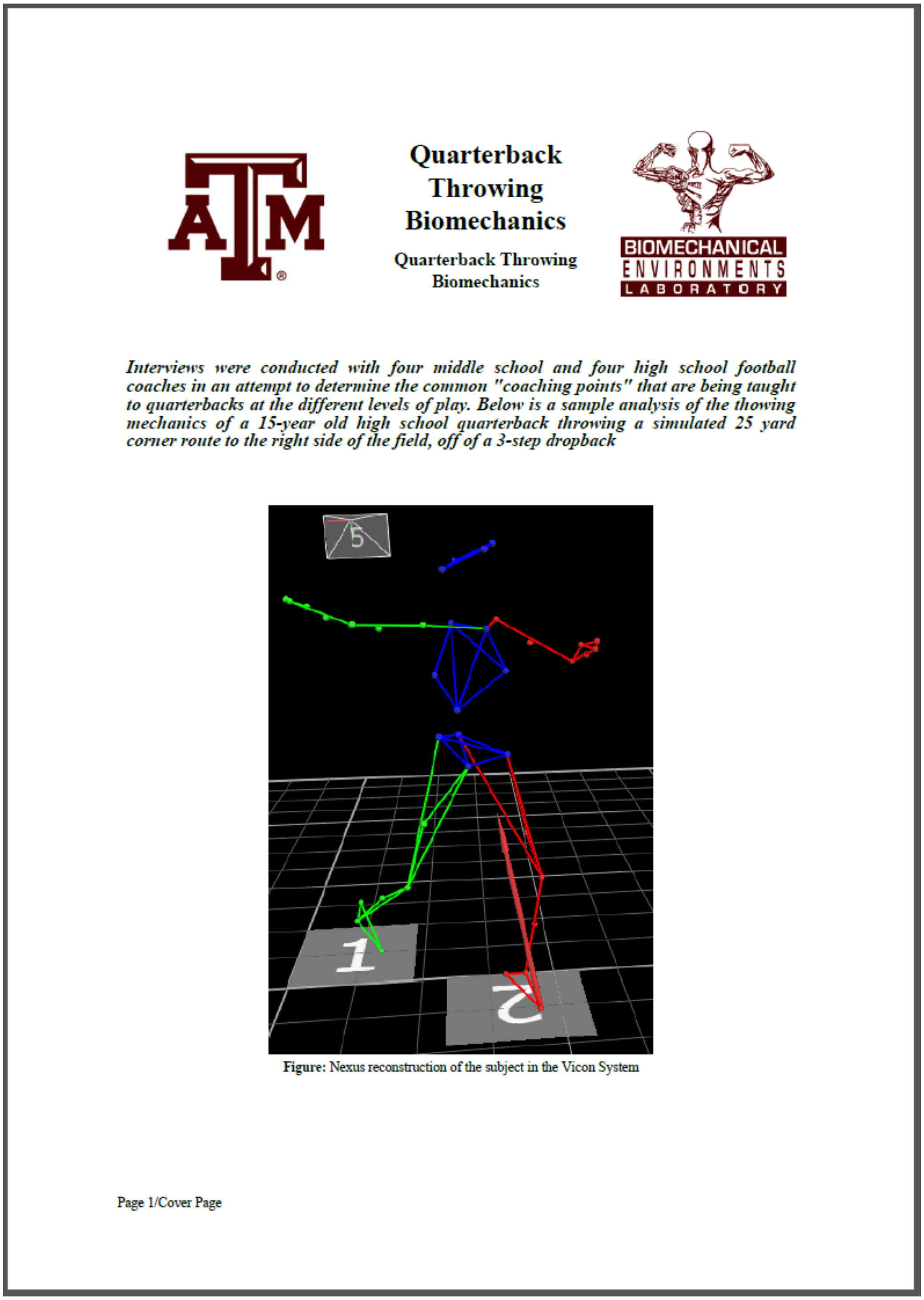


Figure 46: Plot of the swing phase for the right foot of a sheep including the maximum force vector calculated during the foot strike.

5.4 PDF Report Generation

The PDF section of the main UI allows the user of the software to create custom cover pages for their PDF reports of the analysis results, preview them within the UI, and save them to a desired location. The user has the option to select from two different cover page templates and can generate three different report types. The details for the cover page are stored in the project data object so that the user does not have to re-input them each time they generate a report, and can just use the same cover page template for every report. An example cover page is shown in Figure 47 and an example report over an analysis is shown in Figure 48.



Quarterback Throwing Biomechanics

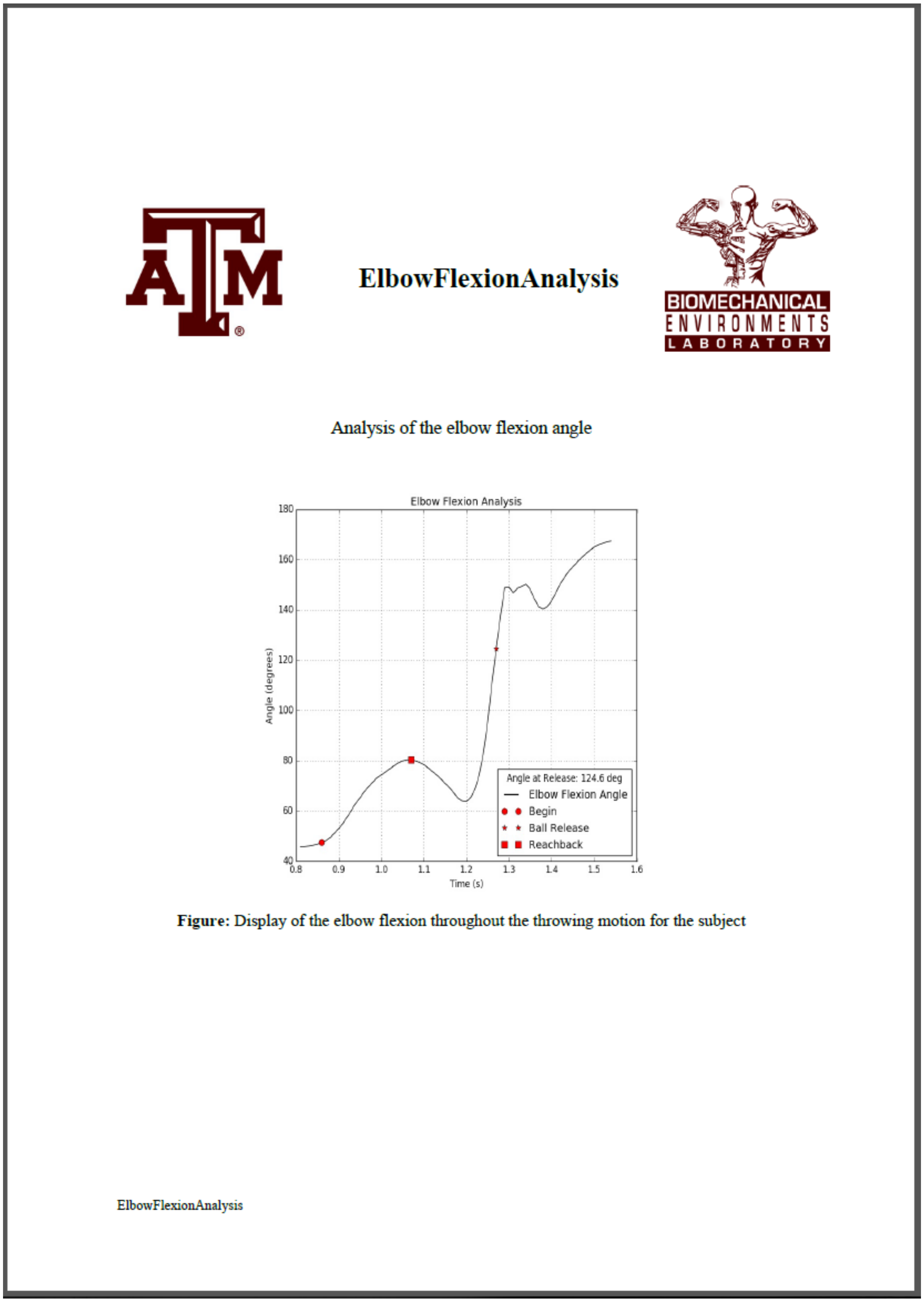
Quarterback Throwing Biomechanics

Interviews were conducted with four middle school and four high school football coaches in an attempt to determine the common "coaching points" that are being taught to quarterbacks at the different levels of play. Below is a sample analysis of the thowing mechanics of a 15-year old high school quarterback throwing a simulated 25 yard corner route to the right side of the field, off of a 3-step dropback

Figure: Nexus reconstruction of the subject in the Vicon System

Page 1/Cover Page

Figure 47: Example cover page for a PDF report for the QB Throwing Biomechanics project.


ElbowFlexionAnalysis

BIOMECHANICAL ENVIRONMENTS LABORATORY

Analysis of the elbow flexion angle

Figure: Display of the elbow flexion throughout the throwing motion for the subject

ElbowFlexionAnalysis

Figure 48: Example of a report page for a specific analysis. The description and the caption are set by the user within the analysis script.

5.5 Discussion

The purpose of performing the analysis of the QB throwing motion was to validate that the software platform provided a flexible framework for programming custom analyses and then displaying those analyses in the user interface and in an example PDF report, while demonstrating all of the different features of the software platform. The software allows for a user to interpret the data in multiple ways. For the QB trials, the coaches sought both numerical and visual interpretations of the results. We used Python as the tool to create the analysis, and utilized the UI to display the visualize results as well as to compile and output the numerical results. The software is not "plug and play" like some commercial software packages, meaning that a user must program their own custom analyses before displaying any results. However, the advantage of this method is that the user is not limited by any 3rd party software capabilities and can create any custom analysis that they wish. As more and more researchers in the lab program analyses for different MOCAP applications we will begin to build up a database of these custom analysis scripts and visualizations that can be used as examples or building blocks to create more advanced analyses in the future. The fact that all of the analysis scripts will be required to follow the same framework for use in the UI will make it even easier to look through the available files to see if there is a useful bit of code that can be repurposed for a new application. This should provide many advantages to an engineering laboratory that is working with different collaborators who may not always want to see the data analyzed in the traditional sense. The results displayed here are all in the form of plots of the data, but a developer can return the data in any form that they wish (arrays, numerical results, strings, etc.).

CHAPTER VI

CONCLUSIONS

6.1 Limitations on Software Package

The major limitation on the software package is that additional functionality or custom analyses have to be programmed using Python, so future users of the platform will have to learn how to program in Python. However, Python uses a very basic syntax and is open source, so there is a vast wealth of information available for developers wanting to learn how to use the language. The c3d module that is currently being used to parse through the c3d data files and access the trajectory and analog data has been modified and only verified to work on Vicon c3d files. While c3d is a standard file format, I have read that some motion capture companies make slight modifications to their c3d outputs and that they aren't all exactly the same. This means that the usefulness of that portion of the package for use with other MOCAP systems will need to be verified using c3d outputs from those systems, and eventually the package would be better served to find a more permanent library to do this data parsing. OpenMA (mentioned in section 2.3.2) is one potential solution to this problem, but that library is not yet completed and available for commercial use. The PDF report generating functionality is fairly limited, and will likely want to be changed on a project to project basis. There is enough PDF generation code available that a user should be able to overwrite the generic functionality and make their own more complex PDF reports fairly easily. The Trajectory Datatype class is functional for the math that was required to perform the analysis that was done over the QB Throwing Mechanics project. There is definitely the need for expansion of that class to include a larger library of mathematical functionality.

6.2 Future Developments

6.2.1 Software Package

Expand Math Functionality

The first step in the future development of the software package is to add new users to the software package and test it out for additional motion capture applications beyond just the QB study. By doing this it will quickly become apparent what math functionality needs to be added to the Trajectory Datatype class to enhance the usefulness of the code. For example, that class currently contains all of the math functionality and error analysis required to perform the football analysis. This includes all of the standard math functions (add, subtract, multiply, etc.), as well as a number of trigonometric and more advanced vector functions (sin, cos, tan, dot and cross product, etc.). However, for a different study, such as the sheep gait analysis, additional math functions may be necessary. One function that has already been brought up is a best fit line function in order to be able to fit a line to the walking path of the sheep. Continuing to build up the library of mathematical functions available to programmers will ensure that this package is useful for all future applications.

Package as an Executable

To allow future users easy access to the software and the ability to use the package without having a Python distribution installed on their computer, the software platform is going to be packaged as an executable. This will allow any user to simply load the program from an .exe file and access the full software functionality, but it will not have to be run from within a Python IDE. In addition to packaging the full framework as an executable package, a subset of the software package will be created for the purpose of distributing to collaborators and allowing them to access the motion capture data, videos, and analysis results for their own viewing and display purposes. This sub package will contain reduced functionality and will not allow collaborators to make any edits to analyses. It will simply be a tool for accessing, viewing, and interacting with the data and results on their own.

Create a Sub-Package of the Platform

A desire has been expressed to create a limited version of the software package and then to make that sub-package available to our collaborators for specific projects. We would then be able to give this package to them, along with the analyzed trial data and video files, and they would be able to use the software to view the video files, and results from that were calculated for individual trials. Which may be of some interest to them.

Update the C3D Framework

As mentioned in the limitations section, the c3d framework that is currently used in the software is not an idea solution and needs to be verified further. The OpenMA framework

[14] mentioned in Chapter 2.3.2 is one potential solution that may work better for the purposes of accessing c3d data. The OpenMA solution is also said to allow users to make edits to c3d data, which may also be a useful future development of this package. Therefore, this different solution needs to be explored, and if it is not feasible, there may be a different solution available, or the current method of parsing c3d data needs to be further verified.

Display Results in Additional Formats

The results from the analysis scripts are returned in a Python dictionary. A dictionary can handle any data type, but currently the results dictionaries only contain strings and matplotlib objects, which are then displayed in the UI. It is likely that future users will want the option to display different types of results in the UI (numerical results, tables, images, etc.). Adding the functionality in the UI to handle the different formats will be trivial. However, a comprehensive list of potential return types needs to be developed so the UI can be programmed to handle all future potential return types and display them in the appropriate way. For example, the first return type that will be added will be a simple floating point value. This will allow us to return key angles, force measurements, and other key measure to the UI and save them for statistical analysis later on. Still, the UI will need to be programmed to know what to do when a floating point value is found and how it should display this return types in the UI as a result of an analysis.

Software Documentation for IP Protection and Potential Open Source Availability

The final software package is almost 10000 lines in length, contains numerous classes, plugins and UIs, and is still growing. The code itself is pretty well documented, but for this package to remain useful long-term, full documentation and instructions on the use of the software will need to be written. There is also the potential for the creation of intellectual property for the software, as well as (if the software is useful enough) the potential to make the software open source and open it up to the biomechanics community for use in other motion capture labs that need a similar tool. However, to pursue any of these paths, the software will first need to be well documented and have an easy-to-follow user manual.

6.2.2 Quarterback Throwing Mechanics

Performing Additional Analyses

The analyses performed so far have been based upon the input received from interviews with local middle school and high school football coaches, and observations made by researchers in the lab. To make this a more robust analysis of QB throwing mechanics, additional analyses will need to be performed based upon the coaching literature and observations into attributes that may affect throwing mechanics. There is also the potential that once the coaches and players see the results of the analysis they will have additional insights into what they wish to see in their results.

Longitudinal Study

A number of the QBs that participated in this study still have years of eligibility left at the middle school and high school levels. One key portion of this study that will differentiate it from any study that has been performed on QB throwing mechanics will be bringing in the same QBs for additional testing and attempting to determine how their mechanics have changed over time. This will give additional insights into how players are developing based upon which system and methods are primarily used to coach their throwing technique.

Analysis of QBs at Higher Levels

Part of the uniqueness of this study was looking at a younger population than has traditionally been used in past studies. However, a useful comparison would be between the throwing technique of a middle school or high school QB and the mechanics of a collegiate or professional QB. This will not only be interesting to see how the mechanics evolve over time, but to see if any similarities can be observed between the advanced QBs and the younger QBs who are most successful or end up continuing on to play at the higher levels.

Results and Conclusion for QB Mechanics

The comparisons that we plan to make from the data collected in this study are: comparison between QBs in different age groups, comparison of QBs trained in different coaching systems, and a comparison of a QB after a year of development and training in a specific coaching system. Based upon the statistical results of these comparisons, we will be able to draw some conclusions about biomechanical differences between QBs at different levels of play and potentially determine what constitutes “good” mechanics and the most effective coaching techniques to employ for a specific age group.

6.3 Conclusion

As a research lab utilizing MOCAP technology for a variety of different applications ranging from medical device research to sports biomechanics and motion optimization, we have the need to analyze data and present results in a variety of different ways all while maintaining continuity between the way the work is done on various projects and by different researchers. There is a large amount of frontend and backend work that goes into accessing and analyzing data for any MOCAP application and oftentimes after the researcher who performed a set of work graduates their work is difficult to user due to a variety of factors. This thesis set out to fill a need in our lab for a data organization and management software platform to synchronize and reduce the amount of effort that is required every time we need to analyze data from our MOCAP system. This was done through establishing a framework which researchers can follow when programming custom analyses of MOCAP data, and creating a user interface which will allow researchers to easily display the results of running their analysis when presenting to collaborators, and also allow for bulk processing of data and compiling and outputting results. Once established for use in our lab, we will begin to build up a database of custom-programmed analysis Python scripts that all were programmed the exact same way and can be used as the building point for researchers working on new applications in the future. Ensuring that researchers stay within the provided framework will also ensure that the work they do is repeatable and reproducible after they graduate and leave the lab. This is especially important for applications like the QB Throwing Biomechanics project, which was used for validation of the software framework created for this thesis, which is a long term longitudinal study and will likely have multiple researchers working on it before it is completed.

This thesis gave a detailed explanation over the inner workings of the software platform and how it will be used for future motion capture applications in our research lab, as well as the background behind the problem, the background behind the QB Throwing Mechanics project, and the future work that will be performed for each of these projects. The analysis performed for over QB throwing biomechanics will become the basis of the analysis for that study and the next researcher will be able to use the software package to bulk process all of the remaining data for that study and compile and output the results. The resulting software package provides a set of tools which researchers can utilize for a wide variety of MOCAP applications by easing the burden of the work that is repeated for all of these projects while still allowing for the flexibility to implement custom analyses for very specific applications.

REFERENCES

1. Rash, G.S. and R. Shapiro, *A Three-dimensional dynamic analysis of the quarterbacks throwing motion in American football.* Journal of Applied Biomechanics, 1995. **11**: p. 443-459.
2. Fleisig, G.S., et al., *Kinematic and Kinetic Comparison Between Baseball Pitching and Football Passing.* Journal of Applied Biomechanics, 1996. **12**: p. 207-224.
3. Kelly, B.T., et al., *Electromyographic analysis and phase definision of the overhead football throw.* American Journal of Sports Medicine, 2002. **30**(6).
4. Heppe, R.A., *The kinematic variables related to the efficiency of throwing: football*, in *Department of Human Performance*. 1992, San Jose State University. p. 104.
5. Wood, J., *The relationship between footwork and the passing performance of high school quarterbacks*, in *Exercise Science*. 2000, Slippery Rock University.
6. Platt, B., *Kinematics and kinetics of two different overhead throws: passing and pitching*. 2012, University of Arkansas.
7. Beeman, A., *A Kinematic and Dynamic Analysis of the American Football Overhead Throwing Motion*. 2015, Rensselaer Polytechnic Institute.
8. Bohnert, K.R., *A complete kinetic, kinematic, and electromyographical analysis of the football throw in collegiate quarterbacks*, in *Kinesiology and Health Promotion*. 2016, University of Kentucky.
9. *The 3D Biomechanics Data Standard*. 2016; Available from: www.c3d.org.

10. C-Motion. Available from: http://www2.c-motion.com/products/visual3d/.

11. C-Motion. *Visual3D Wiki*. 2016; Available from: http://www.c-motion.com/v3dwiki/index.php/Visual3D_Overview.

12. Vicon. *BodyBuilder*. 2016; Available from: https://www.vicon.com/products/software.

13. Barre, A. and S. Armand, *Biomechanical ToolKit: Open-source framework to visualize and process biomechanical data.* Comput Methods Programs Biomed, 2014. **114**(1): p. 80-7.

14. Barre, A. *Open source movement analysis library*. 2016; Available from: http://www.openma.org/.

15. Toolkit, B. *Mokka*. 2016; Available from: http://biomechanical-toolkit.github.io/mokka/.

16. BodyMech. *A MATLAB based open source package for 3D kinematic analysis*. 2006; Available from: http://www.bodymech.nl/.

17. Delp, S.L., et al., *OpenSim: open-source software to create and analyze dynamic Simulations of movement.* Ieee Transactions on Biomedical Engineering, 2007. **54**(11): p. 1940-1950.

18. SimTK. *Welcome to OpenSim*. 2016; Available from: http://simtk-confluence.stanford.edu:8080/display/OpenSim/Welcome+to+OpenSim.

19. Marlbrook. *Biomechanics of Bodies*. 2011; Available from: http://www.marlbrook.com/index.html.

20. AnyBody. *AnyBody Modeling System*. 2016; Available from: http://www.anybodytech.com/.

21. Cappozzo, A., et al., *Surface-marker cluster design criteria for 3-D bone movement reconstruction.* IEEE Trans Biomed Eng, 1997. **44**(12): p. 1165-74.

22. Foundation, P.S. *Python*. Getting Started With Python 2016; Available from: www.python.org.

23. Developers, N. *NumPy*. 2016; Available from: http://www.numpy.org/.

24. Hunter, J., et al. *Matplotlib*. 2014; Available from: http://matplotlib.org/.

25. Ltd., R.C., *PyQt Reference Guide*. 2015.

26. *The Qt Company*. 2016; Available from: https://www.qt.io/.

27. Company, T.Q. *Qt Designer Manual*. 2017 [cited 2017; Available from: http://doc.qt.io/qt-5/qtdesigner-manual.html.

28. ReportLab, *ReportLab PDF Library User Guide*. 2016.

29. McKie, J.X. *PyMuPDF Introduction*. 2016; Available from: http://pythonhosted.org/PyMuPDF/intro.html.

30. Johnson, L. *c3d 0.3.0*. A library for manipulating C3D binary files 2015; Available from: https://pypi.python.org/pypi/c3d.

31. *pickle - Python object serialization*. Data Persistence 2016; Available from: https://docs.python.org/3/library/pickle.html.

32. Pandas. *Pandas: powerful Python data analysis toolkit*. 2017; Available from: http://pandas.pydata.org/pandas-docs/stable/.

33. Company, T.Q. *QTabWidget Class*. Qt Documentation 2016; Available from: http://doc.qt.io/qt-5/qtabwidget.html.

34. Seroyer, S.T., et al., *The kinetic chain in overhand pitching: its potential role for performance enhancement and injury prevention.* Sports Health, 2010. **2**(2): p. 135-46.

APPENDIX A

Full List of Coaching Points

Table 4: Full list of coaching points and the number of times each was mentioned in the interviews.

Coaching Point	**# of Times Mentioned**
3-step drop footwork	4
Feet underneath during quick game	1
Rollout	
release on 4th step	1
45 opening step	2
shoulder orientation	2
arc of rollout	2
Non-throwing arm	
Leading throw, pulling off shoulder through	2
Moves high up to finishing down low/ left arm up	4
Keep it tight	3
Sprint out	1
Nose away / push ball backwards	2
Hips	
Hips stay closed prior to throw	1
Hip lead the throw	3
Open the hips and shoulder up on the throw	2
Shoulders and hips "screwed opposite"	1
Elbow	
90 degree bend in the elbow at the release	1
Hand passes directly over the elbow	1
Elbow leading the hand	4
Wrist above the elbow	1
Keep the elbow up on the throw/above shoulder	2
7 eyes downfield	1
Ball up and over the back shoulder	2
Ball pushed towards back shoulder when holding it	2

Table 4 Continued

Coaching Point	# of times Mentioned
Ball up and ready to go: “between the armpit and the earlobe”	1
Slight lean forward when standing	1
Wrist flick	2
No dip when moving ball backwards	4
Prior to throw -> up on the toes, balance, standing tall	2
Ball parallel to backward motion when coming back	1
"Backwards C" on the reach back	1
Release point can vary based upon the route	3
Follow through to opposite pocket	1
Shoulders stay leveled throughout the throw	1
Shoulders up for deeper throws	1
Very little head movement	1
High release point	1
Back of wrist facing target	1
Thumb and index finger pointing toward the target	1
Lead step is slightly passed the midline of the throw	2
Feet slightly wider than shoulder width	1
Front foot opens up, but doesn't stride much past the front shoulder	1
Big toe push off	1
Toe pointed @ 1 o'clock when they step to throw	1
Weight back, transferred forward on the throw	1
No over stride/ stride length	4
Quick release	5
Back interior malleolus should be facing the target	1
nice medium trajectory on the ball	1
“Camera” on hip and shoulder start off facing the target. “Camera” on the belly faces the target on the release of the ball	3
Drag back foot through, don't lift it off ground	3
Hand vertical above elbow on release	1
For throwing on the run: Moving at the target	1

Methods for Data Analysis

Hand Path

Steps to plot the hand path:

- Access required trajectories
- Using least squares regression fit a line to the forward direction of the hand motion in the xy plane
- Rotate the raw trajectory data by the slope of the your fitted line, using a rotation matrix
- Plot stick figure by connecting each of the rotated trajectories in succession starting with the ankle and ending with the hand
    - Plot this same stick figure at four discrete points in time – beginning of the throw, moment of reach back, moment of ball release, and the end of the throwing motion
- Plot the trajectory of the throwing hand vs time to show the path that the throwing hand took throughout the throwing motion

Hip Leading Angle

Steps to calculate the hip leading angle:

- Create a vector for the hips, and a vector for the shoulder
- Calculate the angle of each vector relative to the global x-axis
- Subtract the relative hip angle from the relative shoulder angle to obtain your "hip leading angle"

- Plot this angle with respect to time to show the change throughout the trial

Example:

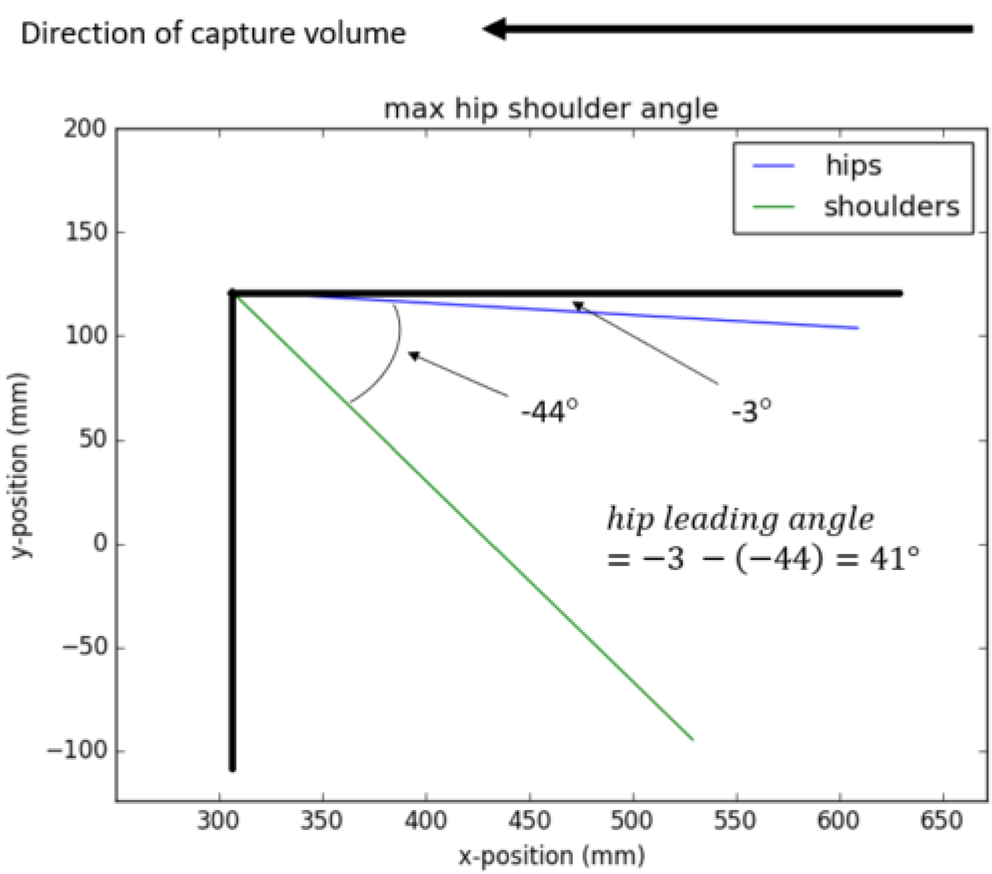


Figure 49: Example of how the hip leading angle is calculated.

Speed Chain

Steps to calculate the speed chain:

- Create a vector for the hips and a vector for the shoulders
- Calculate angle of each relative to the x-axis
- Calculate the rate of change (i.e. the angular velocity) for each angle vector

- Also calculate the rate of change of a marker on the throwing hand (i.e. the velocity of the hand) in order to be able to show the timing of the peak hand velocity
- Plot each of the velocities on the same graph and linearly adjust the hand speed plot so that it is in a scale that can be shown relative to the two angular velocities
    - Be sure to include this linear adjustment in the legend of the plot

Elbow Flexion Angle

Steps to calculate the elbow flexion angle:

- Create vectors for the upper arm (shoulder to elbow) and forearm (elbow to wrist) using the marker locations for each of these points
- Calculate the angle between the two vectors
- Plot the calculated elbow flexion angle vs time

Shoulder Rotation

Steps to calculate the internal/external shoulder rotations:

- Create vectors for the upper arm and forearm segments using the relevant marker locations
- Create a vector from mid-hip to mid-shoulder (Vmhms) using the relevant marker locations
- Establish a local coordinate system at the shoulder:
    - Xu = upper arm segment
    - Yu = Cross product of Xu and Vmhms

  - Zu = Cross product of Xu and Yu
- Project the forearm vector onto the Yu-Zu plane and calculate the angle relative to Yu (this is the internal/external shoulder rotation angle)
- Plot the calculated rotation angle vs time

Orientation at Ball Release

Steps to calculate the release orientation:

- Create vectors for the hips and the shoulders at the instant of ball release
- Take the cross product of each vector and the z-axis to find the normal vector for each of the hips and the shoulders
- Calculate the slope of the resulting normal vectors
- Fit a line to the path of the ball from the point of ball release to the end of the data using a least squares line fitting method
- Compare the slope of the hip and shoulder normal vectors to the slope of the ball path to see how the subject was oriented at the instant of ball release
- Plot the shoulder and hip vectors at the moment of ball release, as well as the normal vectors for each on the same plot as the trajectory of the ball to give a visual reference of the subjects release orientation

Stride Analysis

Steps to calculate the stride parameters:

- Using least squares regression fit a line to the path of the ball from the instant of ball release to the end of the data and calculate the slope of the line
- Fit a line to the locations of the back foot strike and front foot strike, and calculate the slope of the line
- Compare the two slopes to see the direction that the subject stepped compared to the ball trajectory
- Calculate the distance between the two foot strikes to get the length of the stride, normalize by the subjects height for comparison to other subjects

Weight Distribution

- Use the event times of the foot strikes, and the left and right foot markers to determine which force plate each foot strike occurred on
- Access the analog data for the two force plates the subject stepped on
- Calculate the magnitude of the xyz analog data to get the resultant force vector
- Normalize the magnitude by the subjects body weight
- Plot the normalized force data for both force plates and add the key throwing points to the graph