



University of
Massachusetts
Amherst

Application and Evaluation of Lighthouse Technology for Precision Motion Capture

Item Type	Thesis (Open Access)
Authors	Sitole, Soumitra
DOI	10.7275/12615885
Download date	2025-10-18 21:43:03
Link to Item	https://hdl.handle.net/20.500.14394/33760

**Application and Evaluation of Lighthouse Technology for
Precision Motion Capture**

A Thesis Presented

By

SOUMITRA SITOLE

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

September 2018

Mechanical and Industrial Engineering

**Application and Evaluation of Lighthouse Technology for
Precision Motion Capture**

A Thesis Presented

By

SOUMITRA SITOLE

Approved as to style and content by:

Frank C. Sup IV, Chair

Yossi Chait, Member

Andrew K. LaPrè, Member

Sundar Krishnamurty, Department Head
Mechanical and Industrial Engineering

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my parents: Mrunal Sitole and Prashant Sitole, and my family for their unwavering support in all my endeavors.

I would like to express my sincere gratitude to my advisor Frank Sup, for giving me the opportunity to work under his direction at the Mechatronics and Robotics Research Lab. Without his patience, guidance, and support this work would not have been possible.

I am also grateful to FTL Labs Corp. for their assistance and sponsorship for the project. A big thanks to Damon and Jared from FTL Labs for developing a graphical user interface that aided the thesis work.

To my committee members, Andrew LaPrè and Yossi Chait, I am extremely thankful for all the suggestions, critiques and feedback which have played a really important role in shaping the thesis.

Last but not the least, I also wish to thank my lab-mates Ericber, Julio, Mark and Vinh who have helped me stay motivated, and whose help and suggestions have been of great assistance.

ABSTRACT

APPLICATION AND EVALUATION OF LIGHTHOUSE TECHNOLOGY FOR

PRECISION MOTION CAPTURE

SEPTEMBER 2018

SOUMITRA SITOLE

B.E.M.E., UNIVERSITY OF PUNE, INDIA

M.S.M.E., UNIVERSITY OF MASSACHUSETTS, AMHERST, USA

Directed by: Professor Frank C. Sup IV

This thesis presents the development towards a system that can capture and quantify motion for applications in biomechanical and medical fields demanding precision motion tracking using the lighthouse technology. Commercially known as SteamVR tracking, the lighthouse technology is a motion tracking system developed for virtual reality applications that makes use of patterned infrared light sources to highlight trackers (objects embedded with photodiodes) to obtain their pose or spatial position and orientation. Current motion capture systems such as the camera-based motion capture are expensive and not readily available outside of research labs. This thesis provides a case for low-cost motion capture systems. The technology is applied to quantify motion to draw inferences about biomechanics capture and analysis, quantification of gait, and prosthetic alignment. Possible shortcomings for data acquisition using this system for the stated applications have been addressed. The repeatability of the system has been established by determining the standard deviation error for multiple trials based on a motion trajectory using a seven degree-of-freedom robot arm. The accuracy testing for the system is based on cross-validation between the lighthouse technology data and transformations derived using joint angles by developing a forward kinematics model for the robot's end-effector pose. The underlying principle for motion capture using this system is that multiple trackers placed on limb segments allow to record

the position and orientation of the segments in relation to a set global frame. Joint angles between the segments can then be calculated from the recorded positions and orientations of each tracker using inverse kinematics. In this work, inverse kinematics for rigid bodies was based on calculating homogeneous transforms to the individual trackers in the model's reference frame to find the respective Euler angles as well as using the analytical approach to solve for joint variables in terms of known geometric parameters. This work was carried out on a phantom prosthetic limb. A custom application-specific motion tracker was also developed using a hardware development kit which would be further optimized for subsequent studies involving biomechanics motion capture.

CONTENTS

	Page
ACKNOWLEDGEMENTS.....	iii
ABSTRACT	iv
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER	
1 INTRODUCTION.....	1
1.1 Motivation	1
1.2 Goal and Scope.....	5
1.3 Organization	7
2 BACKGROUND.....	8
2.1 Current Prosthetic Alignment Procedure.....	8
2.1.1 Bench Alignment	8
2.1.2 Static Alignment	8
2.1.3 Dynamic Alignment.....	10
2.2 Motion Capture Systems.....	11
2.3 Lighthouse Technology.....	13
2.3.1 System Overview.....	13
2.3.2 System Components	14
2.3.2.1 Base stations.....	14
2.3.2.2 Trackers.....	16
2.3.2.3 Host PC with SteamVR Software	17
2.3.3 Working	18
2.4 Kinematics and Transformations	20
3 APPROACH AND RESULTS.....	29

3.1 Initial Testing	30
3.1.1 COTS testing	30
3.1.2 Software tools acquaintance	30
3.1.3 Determining coordinate frames for tracking.....	31
3.2 System Evaluation.....	32
3.2.1 Data acquisition	32
3.2.2 Data processing for repeatability	36
3.2.3 Alternative data acquisition techniques.....	41
3.2.4 Accuracy testing for the lighthouse system	46
3.2.4.1 Modeling the KUKA robot using forward kinematics.....	48
3.2.4.2 Coordinate frame transformations for cross-validation.....	54
3.2.4.3 Global frame transform between the two systems.....	58
3.2.4.4 Cross-validation between the two systems	60
3.3 Quantification.....	62
3.3.1 Rigid body inverse kinematics approach [T4]	62
3.3.1.1 1R configuration	63
3.3.1.2 2R configuration	66
3.3.1.3 RRP configuration	71
3.3.2 Biomechanics inverse kinematics approach.....	75
3.4 Custom Tracker Development	76
3.4.1 Case and Cover Design.....	76
3.4.2 IMU calibration	78
3.4.3 Sensor placement.....	79
3.4.4 Raw data acquisition	83
3.4.5 Optical sensor recalibration	85
4 DISCUSSION AND FUTURE WORK.....	86
4.1 Custom Tracker Optimization.....	86

4.2 Cross-validation against camera-based motion capture.....	88
4.3 Biomechanics motion capture and gait analysis	89
5 CONCLUSION	94
REFERENCES	95

LIST OF TABLES

Table	Page
Table 1: D-H table for 2R planar manipulator	25
Table 2: Repeatability results from the initial evaluation.....	35
Table 3: Post-processing repeatability results for 4 trials.....	39
Table 4: Post-processing repeatability results for 3 trials.....	39
Table 5: Repeatability results with ground reference tracking.....	44
Table 6: Repeatability results with ground reference tracking for 94 % interval	46
Table 7: Modified D-H table for KUKA LBR iiwa 14 R820.....	48
Table 8: RMS errors between model and experimental data	53
Table 9: RMS errors for system cross-validation.....	62
Table 10: D-H table for the phantom limb (2R case).....	67

LIST OF FIGURES

Figure	Page
1: Static alignment of a lower-limb prosthesis.....	1
2: SPARO Technology – FTL Labs Corp. [12].....	4
3: Theoretical prosthetic alignment.....	9
4: Recommended coronal and sagittal plane alignments using L.A.S.A.R posture [17]	10
5: Dynamic alignment of prosthesis.....	10
6: Camera-based motion capture system	11
7: Gait Analysis using motion capture.....	13
8: Data Flow for lighthouse technology [25].....	14
9: Synchronization Blink and Laser Sweep [25].....	15
10: Deconstructed HTC VIVE lighthouses or base stations [25][26]	16
11: VIVE Tracker and Controller	17
12: Real and virtual world interaction interface using SteamVR	18
13: Sensor Angle Calculation [25].....	19
14: Clock Widths for X and Y directional rotors [25].....	19
15: Transformation math using vector diagram.....	22
16: 2R planar manipulator link labeling diagram	25
17: Elbow up and down configurations for the planar 2R manipulator	27
18: Trigonometric analysis for analytical IK.....	27
19: Base station and tracker coordinate systems	31
20: Lighthouse technology evaluation using a robotic arm	33
21: Repeatability plots for lighthouses mounted high up from initial data acquisition.....	34
22: Refined Y trial post-processing for Low case	37
23: Refined position plots after data processing.....	39

24: Gamma (Z rotation) plot for 4 trials.....	40
25: Setup for reference tracking using an additional ground tracker	42
26: Ground reference tracking transformations using vector diagrams.....	43
27: Repeatability orientation plots with ground reference tracking	45
28: Repeatability position plots with ground reference tracking	45
29: Cross-validation setup using KUKA LBR iiwa 14 R820.....	47
30: Position plots for experimental data and forward kinematics model	52
31: Orientation plots for experimental data and forward kinematics model.....	53
32: Coordinate frames for robot arm's global frame and end-effector	54
33: Tracker placement and their corresponding coordinate frames.....	55
34: Transformed position data for KUKA robot.....	57
35: Transformed orientation data for KUKA robot	58
36: Transformation vector diagram for global frame transform using static trials.....	59
37: Position plots for cross-validation	60
38: Orientation plots for cross-validation	61
39: Setup for phantom limb inverse kinematics.....	64
40: 1R Inverse kinematics testing using transformation arithmetic	64
41: Elbow angle for the phantom limb	66
42: Link labeling and kinematic model of phantom limb.....	67
43: Top view analysis of the phantom arm.....	69
44: Side view analysis of the phantom arm	69
45: Start position for 2R case inverse kinematics.....	70
46: θ_1 and θ_2 for phantom arm (2R case)	71
47: Top view analysis for the RRP configuration	72
48: Side view analysis for the RRP configuration.....	73

49: θ_1 and θ_2 for phantom arm (RRP case)	74
50: Translation 'd' for phantom arm (RRP case)	75
51: Render of CAD model for the case, cover, and assembly	77
52: 3D Printed model and component assembly	78
53: Manual Sensor Placement and sensor visualization tool	81
54: Custom tracker post optical sensor integration	81
55: Simulation results using SteamVR Tracking HDK software for determining tracking performance using 19 sensors	82
56: Gyroscope and Accelerometer raw data for X, Y and Z axes	83
57: Optical sensor raw data for designed tracker	84
58: Real-time tracking using the custom tracker	86
59: Pocket designs to hold photodiodes.....	86
60: Custom tracker position and orientation plots	87
61: Setup for cross-validation against camera-based motion capture	89
62: Setup for biomechanics inverse kinematics.....	91

CHAPTER 1

INTRODUCTION

1.1 Motivation

The Amputee Coalition of America estimates that there are 185,000 new lower extremity amputations each year just within the United States and an estimated population of 2 million American amputees [1] [2]. This number goes to the order of a few million considering the population worldwide. It is projected that the amputee population in the United States will reach 3.6 million by 2050 [3]. For amputees, since the residual limb and the body structure is highly idiosyncratic, every time a lower limb amputee gets a new prosthesis; he/she has to go through the process of aligning it to suit their individual needs and restore the lost functionality effectively.



Figure 1: Static alignment of a lower-limb prosthesis

Prosthetic alignment is a technique of aligning the socket, stump/pylon and the foot elements of a prosthesis with respect to each other such that the effective alignment provides

maximum comfort to the amputee along with a normalized and symmetric gait function. For a transfemoral case, alignment may translate to a few additional components along with the ones mentioned above which are typical for a transtibial prosthesis. The prosthetic alignment technique for a transtibial or a transfemoral lower limb amputee consists of three sequential procedures namely: Bench, Static and Dynamic alignment. Figure 1 shows the general static alignment procedure carried out by a prosthetist. A detailed overview on these techniques has been covered in the Background section (Chapter 2).

Misalignment accounts for issues such as residual limb pain due to uneven stress distribution leading to excessive loading on the proximal or distal regions of the socket, increased metabolic costs implying increased energy expenditure, increased risk of osteoarthritis, back pain, poor balance or instability, irritation, fatigue and reduced mobility [4] [5].

Current prosthetic alignment approach involving the specified alignment techniques is highly iterative and requires a skilled prosthetist to align the prosthesis. The effectiveness of this procedure is a function of the quality of feedback of the fit from the amputee and the nature of the interaction between the amputee and the prosthetist. The methods rely on the prosthetist's trade experience as well as the quality of his/her visual inspection of the amputee's gait dynamics. It is obvious to note that these factors are quite inefficient since an amputee tends to develop compensatory habits overtime that are aberrant to the normal gait and any asymmetry in the motion may not necessarily imply poor or misalignment [6]. Additionally, since the procedure is dependent on the skill of a prosthetist; hence an amputee can have significantly different alignments for the same prosthesis due to lack of a system standard. This lack of a generalized standard not only accounts for inconsistencies in the alignment; but it may also lead to issues pertaining to misalignment.

As opposed to current abstractions, Boone, Kobayashi, and others have established through their studies that there are several mathematical and tangible factors that can be measured for indicating issues in alignment [7] [8]. Socket reaction moments and ground reaction forces have been found to be some of the notable factors that indicate misalignment [9]. Other studies have established that radiographic parameters and gait or temporal parameters can also serve as important factors for guiding alignment [10] [11]. Hence, instead of using a system that relies on parameters such as visual inspection, experience, and skill of the prosthetist, more pronounced and effective methods can be devised to capture motion and align a prosthesis.

A potential way of capturing motion in this respect as well as for quantifying prosthetic alignment is using the camera-based motion capture. However, these commercial systems are highly expensive and inaccessible outside high end research laboratories. This calls for a need for a motion capture technology that can be scalable, easily accessible and most importantly, cost-effective without compromising the system accuracy and repeatability.

Sensorized Prosthetic Alignment Readout (SPARO) technology proposed by FTL Labs Corporation is one such system which leverages the lighthouse technology used for motion tracking in virtual reality systems. The SPARO technology which has been in development in collaboration with University of Massachusetts Amherst since July 2017 aims at quantifying prosthetic alignment. The objective was to develop an effective alignment capture system that can be easily accessible to the prosthetists, which would provide real-time feedback on the desired alignment thus minimizing the human factors involved and associated discrepancies in the loop. The system would provide a record of the past alignments along with a visual tool to see in practice the results of the alignment adjustments that they try out as well as a feedback application for the amputee where he/she can log the alignment quality over time. Figure 2 portrays how the system would look post development.

This thesis work augments FTL's work with suitable contributions. FTL has already developed a virtual interface using Unity engine with the SteamVR plugin (a support tool that allows interfacing with the VR hardware components). The interface will be used for data acquisition and to visualize the performance for this thesis work.

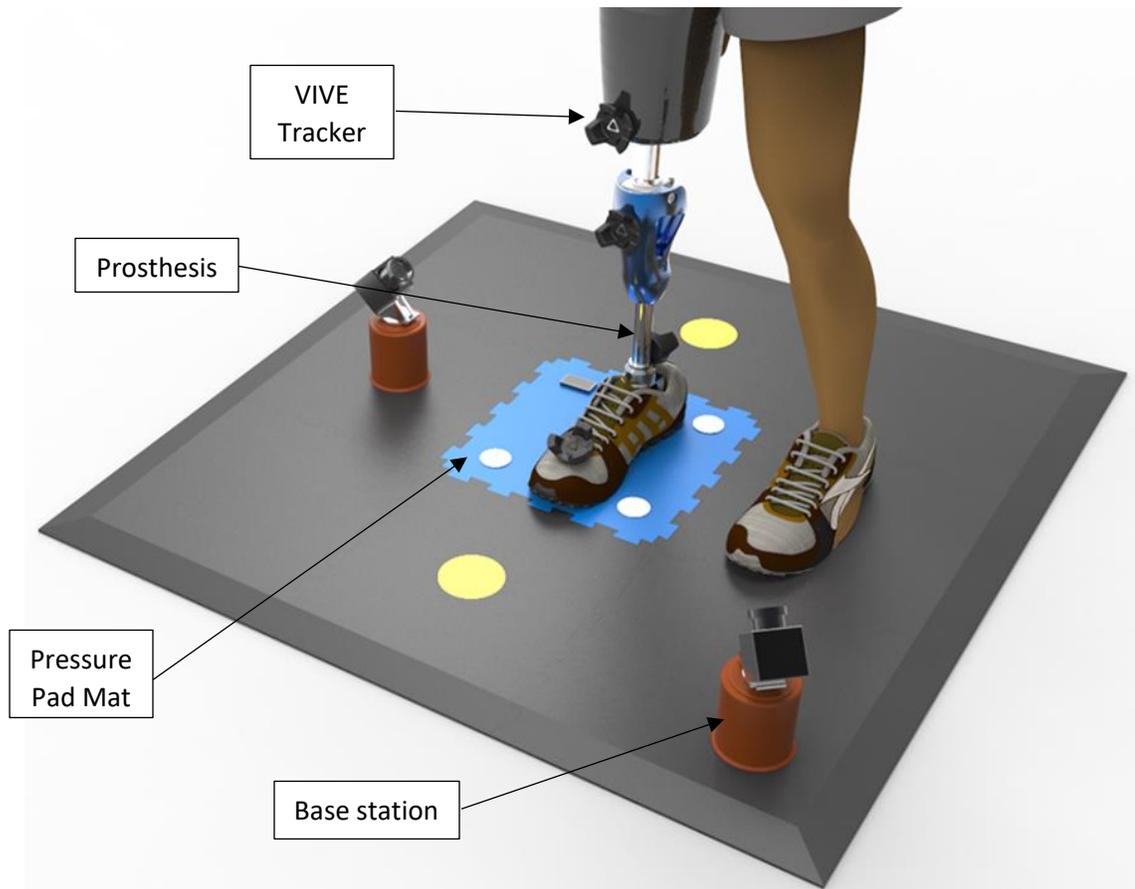


Figure 2: SPARO Technology – FTL Labs Corp. [12]

1.2 Goal and Scope

This thesis work aims at providing a low-cost solution for motion tracking applications in medical and biomechanical fields demanding precision motion capture such as gait analysis by leveraging the lighthouse technology (SteamVR Tracking) as opposed to the current commercially available technologies. The project also serves as an extension to the sensorized prosthetic alignment readout technology proposed by FTL Labs, Amherst in collaboration with Mechatronics and Robotics Research Lab at the University of Massachusetts Amherst. This system aims to capture prosthetic alignment. Rather than relying on trade experience and skill of a prosthetist, it proposes to quantify alignment. The thesis work aims to augment FTL's proposed technology in terms of system evaluation, custom tracker development and inverse kinematics, and draws out further implications for extending the system.

The lighthouse technology has been developed and proven for motion tracking for virtual reality systems. In the current industry, the technology has not been broadly applied outside virtual reality applications. This thesis work aims at addressing any shortcomings in terms of accuracy, repeatability, and scalability to further develop motion capture for medical and biomechanical applications using the lighthouse technology. Current motion capture technologies for gait analysis involve the use of camera-based systems along with force plates and post processing softwares which offer sub-millimeter level accuracy. However, they cost about a couple hundred thousand dollars. The proposed system uses patterned infrared light emitters and objects embedded with photodiodes which give out a digital signal when hit with the infrared light. Paired with a pressure pad mat or footplate for ground contact profile; the system would cost about 2-3k USD which is significantly less than the current commercial video-based motion capture solution [13].

The implication of the use of lighthouse technology for precision motion tracking is not limited to the biomechanical or medical fields. Provided that the system gives satisfactory results in terms of accuracy, repeatability and other performance parameters; this technology can be scaled to other applications in science, engineering and robotics. The technology can be used for precisely locating manipulators, teaching robotic paths, tele-operating rescue robots, locating and controlling robotic swarms like drones, controlling collaborative robots and for several other applications.

1.3 Organization

The 'Introduction' section (Chapter 1) begins with the motivation behind the thesis work highlighting the significance, goals and the scope of the proposed work. The 'Background' section (Chapter 2) reads through the current methods and technologies used for the mentioned applications. This is followed by the overview of the lighthouse technology; and the various components used. The fundamentals of kinematics used to capture and quantify motion for this work also have been mentioned here. Three main objectives were planned: initial testing, system evaluation based on repeatability and accuracy criteria and, quantification. The subsequent chapter on 'Approach and results' (Chapter 3) documents the objectives, tasks, methods used, and their corresponding results. The subsequent chapter (Chapter 4) goes through the implications of the results and the future work. The report concludes with Chapter 5 highlighting important results.

CHAPTER 2

BACKGROUND

2.1 Current Prosthetic Alignment Procedure

An overview of the current sequential procedure that a prosthetist relies on for aligning a prosthesis is as follows:

2.1.1 Bench Alignment

This alignment technique as the name suggests is carried out on a bench by a prosthetist without the amputee wearing the prosthesis. The prosthetist uses a vertical plumb line passing through the socket and center of the ankle bolt or heel of the shoe to align the prosthesis visually [14] [15]. An assistance jig can be used to facilitate alignment [16]. The alignment screws are set to neutral positions so that corresponding changes can be made using further procedures involving static and dynamic alignments.

2.1.2 Static Alignment

In this alignment technique, the prosthetist aligns the prosthesis by taking into consideration the weight distribution and the alignment load line. Figure 3 shows the alignment load line which determines the theoretical alignment. The amputee is made to sit or stand wearing the prosthesis, and suitable adjustments are carried out by the prosthetist to correct the load related factors and improve static load bearing. The primary objective is that the amputee should be able to balance his/her body effectively while standing. Socket height and orientation are set to the optimum to improve weight bearing. Optimum height can be determined by pelvic level placed on the iliac crest with even pressure. Assistive tools can be used to facilitate the alignment process. One such tool is Ottobock's L.A.S.A.R posture which projects a harmless reference laser line to indicate the load line on the amputee's body. The load line should be centered up to 2 cm laterally from the knee joint in the coronal plane, and

the prosthesis should bear at least 40% of the user's body weight. The display and control unit indicate weight and provide an objective tool to determine the distance between the center of pressure and knee axis [17]. For the sagittal plane adjustment, the load line should lie 30 mm anterior to the rotation point of the knee. Figure 4 shows the recommended adjustments in the two planes suggested by Ottobock using L.A.S.A.R posture. Adjustments can then be made based on comfort parameter for the amputee, and the permitted offset is about 15 mm between the comfort and theoretical alignment.

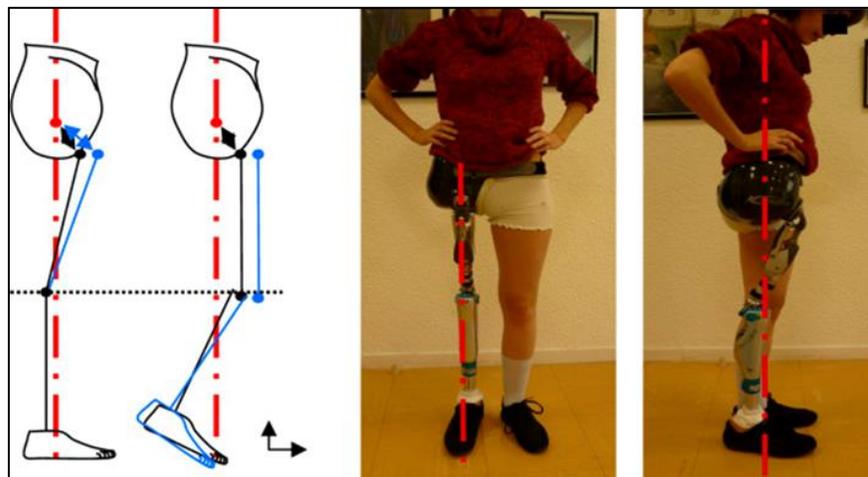


Figure 3: Theoretical prosthetic alignment

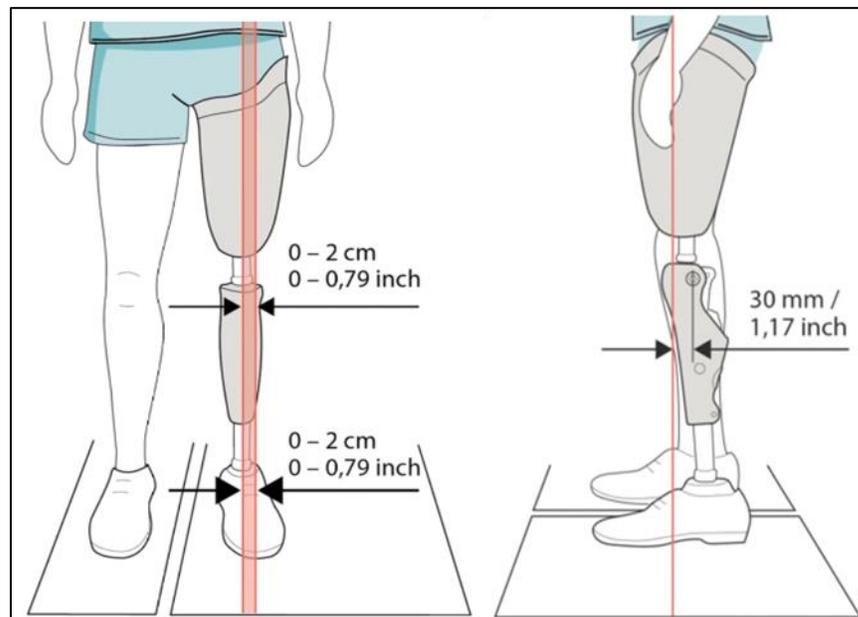


Figure 4: Recommended coronal and sagittal plane alignments using L.A.S.A.R posture [17]

2.1.3 Dynamic Alignment

For dynamic alignment, the amputee is made to walk along a parallel bar, and the prosthetist watches and analyses the gait for any deviations or asymmetries. Figure 5 shows this scenario. Various factors are observed, and progressive adjustments are made based solely on visual inspection in coronal and sagittal planes. Some of the important gait deviations that are observed in the coronal plane are: pylon positions, lateral thrusting, limb pistoning (recommended value should be less than 6 mm). For sagittal plane, following factors are observed to determine gait deviation: knee flexion at heel strike and midstance, drop-off (feeling of falling) rate and plantar flexion. Simultaneously, suitable feedback is obtained from the amputee about the comfort, fit and mediolateral stability of the prosthesis based on their perception [18]. The procedure continues iteratively until both the patient and the prosthetist agree that further adjustments will no longer produce any noticeable improvement. Prosthetist's skill and experience are important factors that drive this technique.



Figure 5: Dynamic alignment of prosthesis

2.2 Motion Capture Systems

An effective way of capturing gait and quantifying joint kinematics is using motion capture systems in a gait lab. These systems may also enable to precisely quantify prosthetic alignment; but due to possible reasons stated in the subsequent sections, they don't seem to be a viable alternative.

The camera-based systems rely on methods which make use of a technique called stereophotogrammetry which involves acquiring multiple two-dimensional images of the body to be tracked from various vantage points using cameras and superimposing the images thus obtained to have a three-dimensional motion file using post-processing softwares and PnP algorithms [19][20]. Modern systems also enable to capture a large number of frames every second to obtain similar results. Figure 6 shows a camera-based motion capture system.

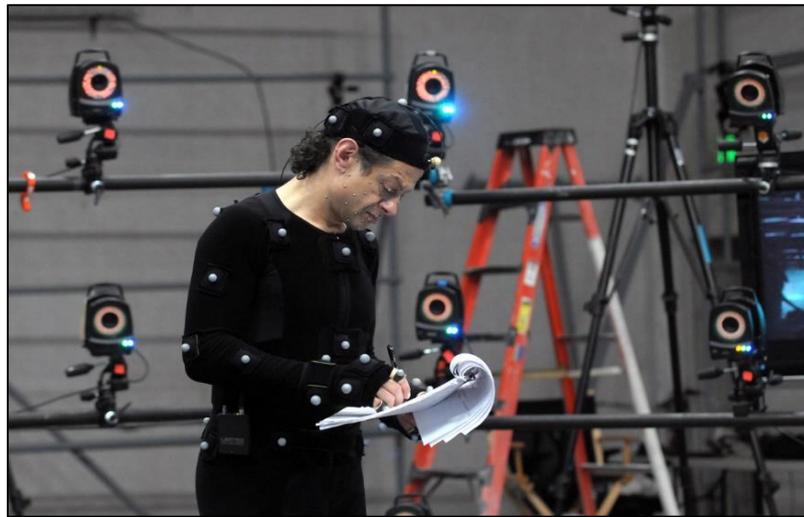


Figure 6: Camera-based motion capture system

For gait analysis, a musculoskeletal model can be obtained by placing passive or active markers on bony body prominences or anatomical landmarks to localize joints using regression models and then using interpolation techniques to construct the connecting links

[21] [22]. The stated marker-based system can be passive or active. Figure 7 shows an active marker-based system used for gait analysis and Figure 6 highlights the use of passive markers. An active system produces its own luminescence by using a series of LEDs and needs to be powered throughout sampling. Active marker systems have the following advantages: low marker jitter due to high signal-to-noise ratio, high resolution of about 0.1 mm. More commonly used are the passive markers which are coated with retroreflective materials. For these markers, unlike the active systems; external light sources are required to highlight them which reflect the incident light. High-contrast thresholding and similar methods are then used for image acquisition. Additional methods can also be used to reduce skin and tissue artifacts to improve accuracy [21]. The motion tracking procedure using passive markers has been found to be sub-millimeter level accurate [23]. Additionally, during analysis, force plates are crucial for identifying gait events precisely. They determine the center of pressure and ground reaction forces as well as the ground contact profile. Combined with a body model, joint moments and power can be calculated automatically using gait analysis software. Though this camera-based motion acquisition system is highly accurate, it is highly expensive and has limited accessibility. Additionally, the user requires extensive training to use such systems due to which this technology has been limited to state of the art research facilities.

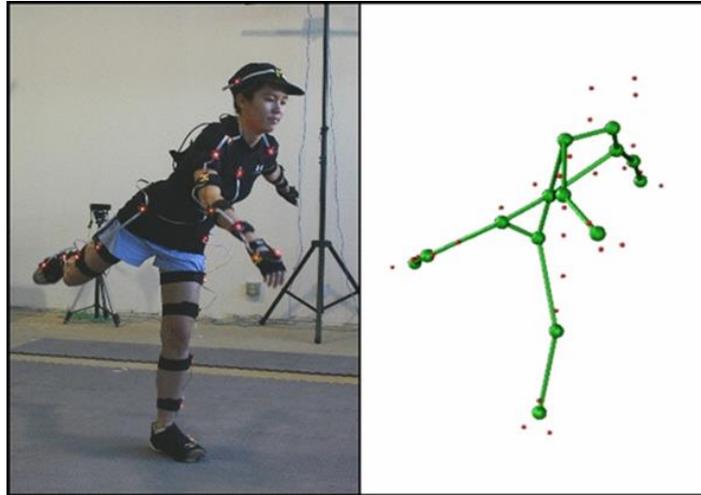


Figure 7: Gait Analysis using motion capture

2.3 Lighthouse Technology

2.3.1 System Overview

As opposed to scrupulously looking at images through a head-mounted display; current leaps in motion tracking systems enable individuals to interact with the virtual environment by performing physical actions or motions in the real world and simulating these movements in the virtual world thus escalating the experience to a whole new level.

The lighthouse technology seems promising in regard to providing a cost-effective motion tracking solution. Alternately known as SteamVR Tracking; this technology is a motion tracking or motion capture system used by HTC for their VIVE Virtual Reality systems. It embodies a position and orientation tracking system with high resolution, speed, and sub-millimeter level accuracy [24]. The absolute position is acquired by triangulation using optical sensors integrated inside the object to be tracked which detect infrared (IR) reference signals sent from a base station to the object embedded with optical sensors or photodiodes. The arrival of reference signals at each sensor is timestamped using a 32-bit counter running at 48 MHz. An onboard IMU augments position and orientation data at a higher frequency rate (250 - 1000 Hz) to update the optical data, and in case of an occlusion, data can be

obtained purely using the IMU based on acceleration and gyroscope. The system is quite analogous to the geo-positional satellite system where multiple satellites triangulate the exact position of the device or the GPS receiver and in case of an occlusion, predicted data is obtained using an IMU or using other dead reckoning techniques.

Data flow in the lighthouse technology takes place in the same order as mentioned. The base stations' synchronization pulses and laser sweeps are intercepted by the photodiodes which send the data to the onboard microcontroller. This information is augmented with the IMU data to increase accuracy. The data is then transferred to the host computer with SteamVR software that determines the tracker's position and orientation. The flow of information is indicated in Figure 8:

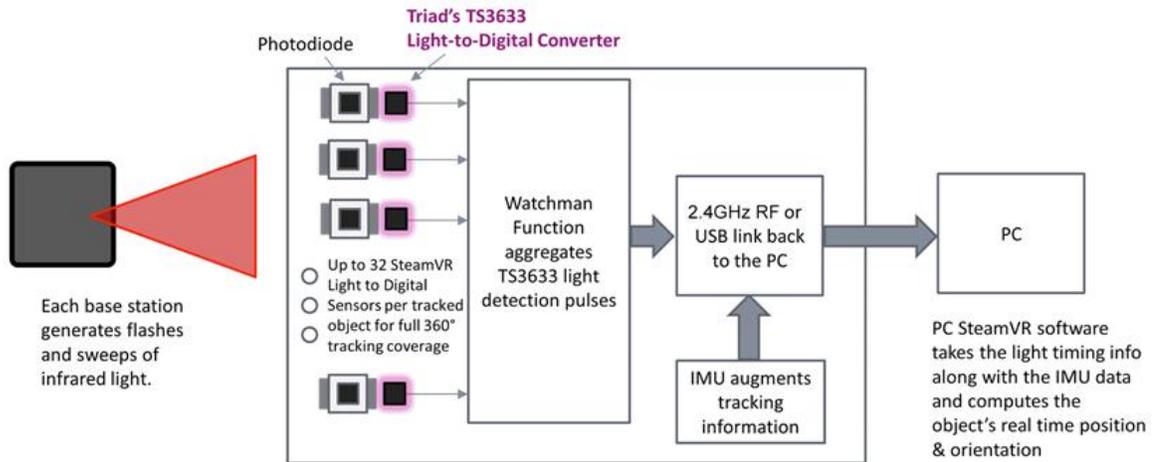


Figure 8: Data Flow for lighthouse technology [25]

2.3.2 System Components

The system makes use of the following components:

2.3.2.1 Base stations

Base stations commonly known as lighthouses are essentially light emitting stations which emit patterned infrared light of 830 nm wavelength. Inside a base station are a series of high powered infrared LEDs known as the sync blinkers which produce a global flash which is a

synchronization pulse that indicates the start of a frame. Additionally, there are two rotors mounted orthogonally operating with a phase offset of 180° which send infrared light sweeps in X and Y axes respectively. The base stations have mirrors which deflect the laser beams sending them out through a fan lens as a laser line which is parallel to the axis of rotation of the base station. Figure 9 shows the global flash and the laser sweeps produced by the base station by the series of infrared LEDs and the rotors.

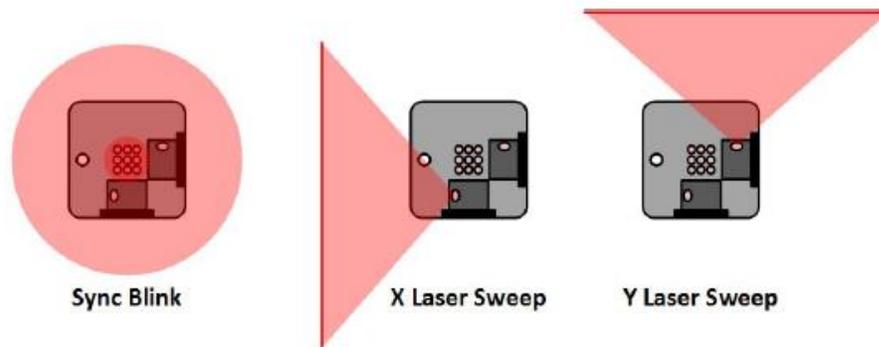


Figure 9: Synchronization Blink and Laser Sweep [25]

The aperture of laser sweeps ranges to about 120 degrees for every rotor due to the fan lens. Data acquisition frequency is about 60 Hz determined by the laser light sweeps using a single base station. However, the usual configuration involves the use of multiple base stations since that allows for more coverage and increased accuracy. In case of 2 base-stations, one base station behaves as a master and the second serves as a slave. The trackable volume using such a configuration is about 125 cu. m. The acquisition frequency drops down to half, since the base stations would then alternate for the sweeps. Multiple base stations also cater for occlusion related issues. Figure 10 shows the lighthouse rotor construction with the fan lens and the mirror. The right-side figure shows the partly disassembled model with the series of LEDs and the laser sweeping rotors.

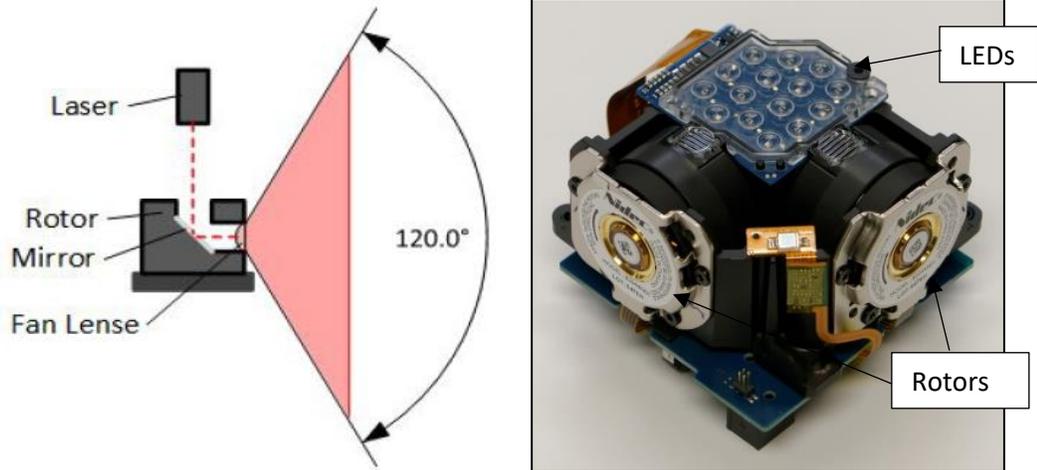


Figure 10: Deconstructed HTC VIVE lighthouses or base stations [25][26]

2.3.2.2 Trackers

Trackers are objects embedded with optical sensors. The optical sensors are essentially photodiodes which function as a light to digital converters. Whenever a photodiode gets hit with a laser sweep or synchronization pulse from the base station it produces a current by generating electron-hole pairs due to photon absorption. The current is amplified using a trans-impedance amplifier into a voltage which then undergoes full-wave rectification and envelope sampling to record the timestamp of when it gets hit. The timestamp is generated using 32-bit counters operating at 48 MHz, and three parameters are sent to the onboard FPGA. The sensors send the synchronization pulse timestamp, the laser sweep timestamp as well as their individual identification number called the sensor channel number data to the application board. The optical data is augmented with the IMU data for improved accuracy. The onboard IMU samples at 250 to 1000 Hz based on wireless or wired communication. The IMU also performs dead reckoning tasks in case the optical sensors lose sight to the base stations. Using the IMU, the orientation then can be obtained from the gyroscope, and the position can be obtained from the accelerometer. The microcontroller then sends this data to the PC with SteamVR either wirelessly or over USB for triangulation. The appropriate

placement of the photodiodes on the tracker is critical since that is what determines the overall tracking effectiveness and system accuracy.



Figure 11: VIVE Tracker and Controller

2.3.2.3 Host PC with SteamVR Software

SteamVR software does the position and orientation triangulation from the time stamp data obtained from the trackers. The detailed procedure for this calculation has been documented in the subsequent section. To evaluate how the tracking translates from the physical world to the virtual world; SteamVR plugin can be used on a computer running the Unity engine to visualize the motion. Figure 12 shows the Unity environment that uses SteamVR plugin to visualize and track objects.



Figure 12: Real and virtual world interaction interface using SteamVR

2.3.3 Working

The angle a sensor makes with the base station that determines the bearing of the tracker relative to the base station is obtained by the following relation [25]:

$$\theta = t \cdot T_{counter} \cdot \omega_{rotor} \text{ rad}$$

Where,

$t = t_1 - t_0$ clock ticks and,

t_1 = laser sweep clock tick for the sensor

t_0 = synchronization blink clock tick for the sensor

$T_{counter}$ is determined by the counter frequency $f_{counter}$ which samples at 48 MHz :

$$T_{counter} = \frac{1}{f_{counter}} = \frac{1}{48e(-6)} \text{ s}$$

The laser sweeps occur at 60 Hz determined by the rotor frequency. Hence,

$$\omega_{rotor} = \frac{120\pi}{1} \text{ rad/s}$$

Therefore,

$$\theta = t \cdot \frac{1}{48e(-6)} \cdot \frac{120\pi}{1} \text{ rad}$$

$$\theta = t \frac{\pi}{400000} \text{ rad}$$

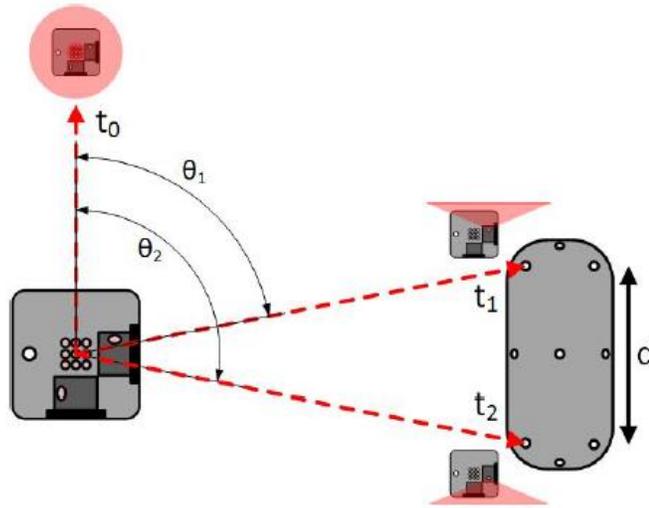


Figure 13: Sensor Angle Calculation [25]

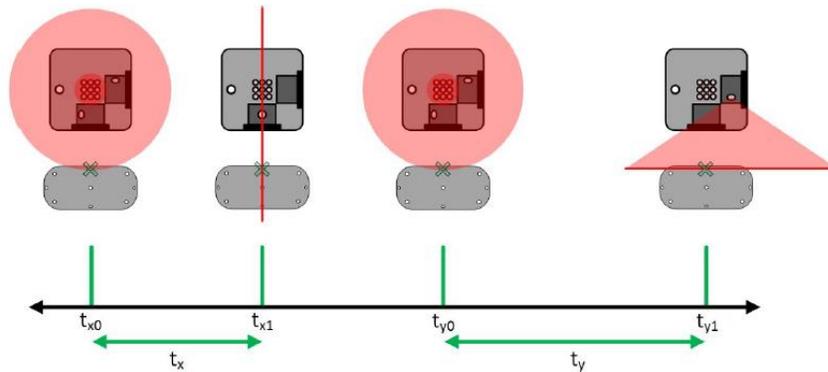


Figure 14: Clock Widths for X and Y directional rotors [25]

As it turns out that the only parameter that is required to determine the bearing to the base station for a tracker is the width of the clock tick that is the difference between the reference synchronization blink and the laser sweep clock tick for that particular sensor. For a base

station, as shown in Figure 14 the clock width is calculated in X and Y direction since one rotor sweeps in X and the other rotor sweeps in the Y direction.

SteamVR then solves for translation and rotation based on the known geometry of the trackable object as well as the location of the sensors which are encoded in a JSON (javascript object notation file). It is important to note, however, that SteamVR requires 5 sensors to be highlighted to determine a pose. In other words, since SteamVR fits in the known geometry and angles for calculating the position and orientation; in case less than 4 sensors are highlighted, it can find multiple sensor combinations that can satisfy the measured angles and geometry, and will not yield a unique position. Theoretically, 4 sensors are required to be seen by the base station for any orientation of the trackable object, since a unique plane can be constructed using 3 points, and an offset the fourth sensor which lies at least 8 mm away from the three planar sensors localizes the geometry. Practical considerations suggest having 5 sensors visible to the base station as a safety factor to avoid object boot up issues.

Augmentation with an IMU provides data at a higher frequency along with dead reckoning. It is important to note that to obtain a position from acceleration one has to integrate twice in time and any errors in acceleration accumulate linearly in velocity and quadratically in position approximation. This induces a lot of drift in the position data which escalates with time which makes it impossible to use the IMU for accurate position determination just by itself. Hence, it is important to design the tracker such that a minimum number of sensors are always visible to the base station and the system does not have to solely rely on IMU data.

2.4 Kinematics and Transformations

The quantification of motion using this technology has been carried out using forward and inverse kinematics. In robotics or biomechanics, kinematics deals with motion studies by the virtue of the configuration of a body relying on geometrical or anthropometric parameters

without considering the cause and external effects such as forces or masses causing the motion. Kinematics can be divided into two types:

- i) Forward kinematics: This deals with finding the end-effector pose when the geometrical parameters as well as joint variables such as joint angles, joint translations, etc. are known.
- ii) Inverse kinematics: This is associated with determining the joint variables when the end-effector pose is known.

Transformations between different coordinate frames express the translations and rotations of one frame with respect to the other. These geometrical transformations mostly pertaining to coordinate frames play an important role in kinematics. Homogeneous transformations are a versatile way to express these transformations [27]. In short, a homogeneous transformation which forms a part of the special Euclidian or Lie group (SE3) encompasses the translation and rotation between two coordinate frames in a 4x4 matrix. Of this matrix; the first 3x3 matrix is the rotation matrix belonging to the special orthogonal group (SO3). Since, these matrices form a class of this group, various properties such as closure, transpose equivalent to inverse or other identities can be used to aid transformations [27]. The three elements of the last column of the homogeneous transformation matrix are positions or rather translations in x, y and z directions respectively for translating the (i-1)th frame to (i)th frame. The last row of the homogeneous transformation matrix is for book keeping purposes and is constant. The transformation matrix can be written as:

$$T_i^{i-1} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where, the rotation matrix is:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

And the position vector is:

$$P = [x \quad y \quad z]^T$$

Consider for instance, we have two points in space P and Q, and their positions and orientations are known with respect to a common point in space say O where, the coordinate frames for each follow the Cartesian coordinate system. The transformations can then be expressed as vectors as shown in Figure 15. The figure does not show the individual frames for convenience of representation. However, each point or vertex of the vector triangle will have a coordinate system independent of the other. In order to express the position and orientation of the point Q in point P's coordinate frame, we can then follow the vector resultant diagram; but unlike addition for vectors, this applies to corresponding multiplications for the transformations.

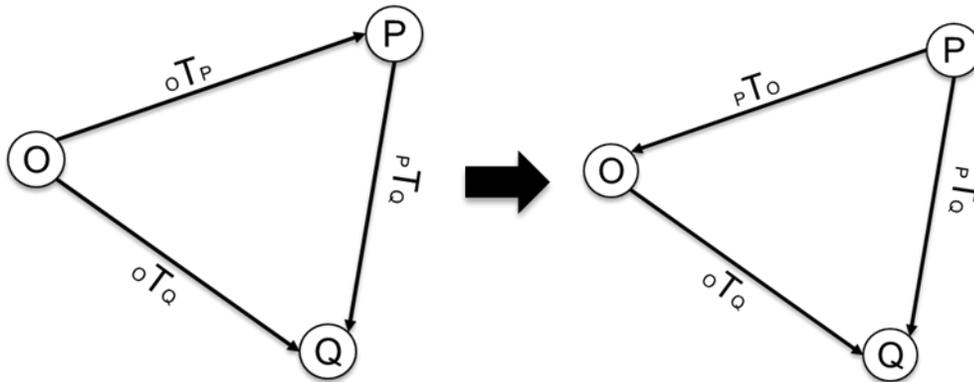


Figure 15: Transformation math using vector diagram

T_P^O and T_Q^O are known transformations. In order to find T_Q^P ; we need to reconstruct the diagram by expressing the unknown T_Q^P as a resultant in terms of the known transformations.

Hence, we need to calculate T_O^P ; which is the inverse of the transformation T_P^O . This is again based on the properties of the homogeneous transformation matrices [27]. The final transform is the product of this inverse transform and the transform T_Q^O .

$$T_Q^P = (T_P^O)^{-1} \cdot (T_Q^O) = (T_O^P) \cdot (T_Q^O)$$

It is important in kinematics that the frame labels or link labels follow a particular convention. One such approach is the Denavit-Hartenberg (D-H) method which is commonly used for forward kinematics [28]. The D-H approach reduces system redundancies by capturing any 3-dimensional transform using just 4 parameters: 2 for translation (a, d) and 2 for rotation (θ , α) instead of 6 parameters describing 3 translations and 3 rotations along each axis. There are 4 rules that need to be followed to layout coordinate frames using this convention [29][30]:

- i) Z_i axis is the axis of rotation and axis of translation for the joint for link 'i'.
- ii) X_i axis should be perpendicular to Z_i and Z_{i-1} axis
- iii) The Y_i frame should be assigned using the right handed coordinate system
- iv) Each X_i axis should intersect the previous joint's Z axis or the Z_{i-1} axis

Once the link labeling is complete, the D-H table can be formulated by finding the 4 D-H parameters from the link diagram.

θ_i is the angle between the X_{i-1} and X_i axes where the rotation takes place about Z_{i-1}

α_i is the twist angle between Z_{i-1} and Z_i axes where the rotation takes place about X_i

a_i is the distance between the centers of 2 coordinate frames along axis X_i

d_i is the distance between the centers of 2 coordinate frames along axis Z_{i-1}

The individual transformation matrices can be further calculated based on the following order of transformations:

$$T_i^{i-1} = [Transn(d_i)]_{z_{i-1}} \cdot [Rotn(\theta_i)]_{z_{i-1}} \cdot [Transn(a_i)]_{x_i} \cdot [Rotn(\alpha_i)]_{x_i}$$

Based on this sequence we get the following transformation matrix:

$$T_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The analytical inverse kinematics method does not have a generalized technique. It essentially involves solving the geometry of the model by using trigonometric approaches to express desired joint variables in terms of known geometric parameters as well as the end-effector position coordinates. Here, an example of a 2R planar manipulator is provided to explain this approach along with forward kinematics. Often forward kinematics can yield equations that can simplify the analytical inverse kinematic solution.

Objective: From Figure 16 find: θ_1 and θ_2 for 2R arm

Given:

- a) Model parameters: l_1, l_2
- b) End-effector or final frame position: X_2^0, Y_2^0, Z_2^0

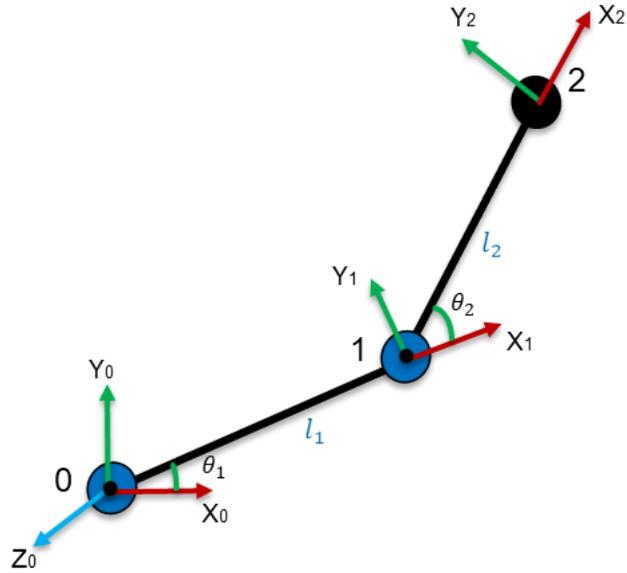


Figure 16: 2R planar manipulator link labeling diagram

Forward Kinematics:

Standard DH Homogenous Transformation matrix:

$$T_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Table 1: D-H table for 2R planar manipulator

Link (i)	θ_i	α_i	a_i	d_i
1	θ_1	0	l_1	0
2	θ_2	0	l_2	0

$$T_1^0 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & l_1 c\theta_1 \\ s\theta_1 & c\theta_1 & 0 & l_1 s\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & l_2 c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & l_2 s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^0 = T_1^0 \cdot T_2^1$$

$$T_2^0 = \begin{bmatrix} c\theta_{12} & -s\theta_{12} & 0 & l_1 c\theta_1 + l_2 c\theta_{12} \\ s\theta_{12} & c\theta_{12} & 0 & l_1 s\theta_1 + l_2 s\theta_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$X_2^0 = l_1 c\theta_1 + l_2 c\theta_{12}$$

$$Y_2^0 = l_1 s\theta_1 + l_2 s\theta_{12}$$

Inverse Kinematics:

From the forward kinematics relation, we have;

$$(X_2^0)^2 + (Y_2^0)^2 = l_1^2 + l_2^2 + 2l_1 l_2 c\theta_2$$

$$\theta_2 = \cos^{-1} \left[\frac{(X_2^0)^2 + (Y_2^0)^2 - l_1^2 - l_2^2}{2l_1 l_2} \right]$$

Generally, the usage of arcsines and arccosine is avoided since they lead to inaccuracies. Using the half angle formula, we have:

$$\theta_2 = \pm 2 \operatorname{atan2} \sqrt{\frac{(l_1 + l_2)^2 - [(X_2^0)^2 + (Y_2^0)^2]}{[(X_2^0)^2 + (Y_2^0)^2] - (l_1 - l_2)^2}}$$

The two values for θ_2 are due to the elbow up and elbow down configuration. These are kinematic redundancies and can be avoided by imposing joint constraints such as: (θ_1 & $\theta_2 > 0$)

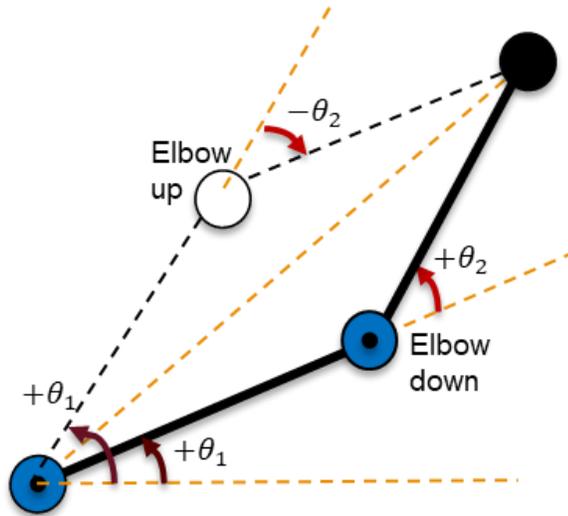


Figure 17: Elbow up and down configurations for the planar 2R manipulator

The geometric approach relies on constructing triangles and applying trigonometric relations for finding the equations of the desired values. θ_2 is now known from the forward kinematics relation. For finding θ_1 , we construct the following triangles:

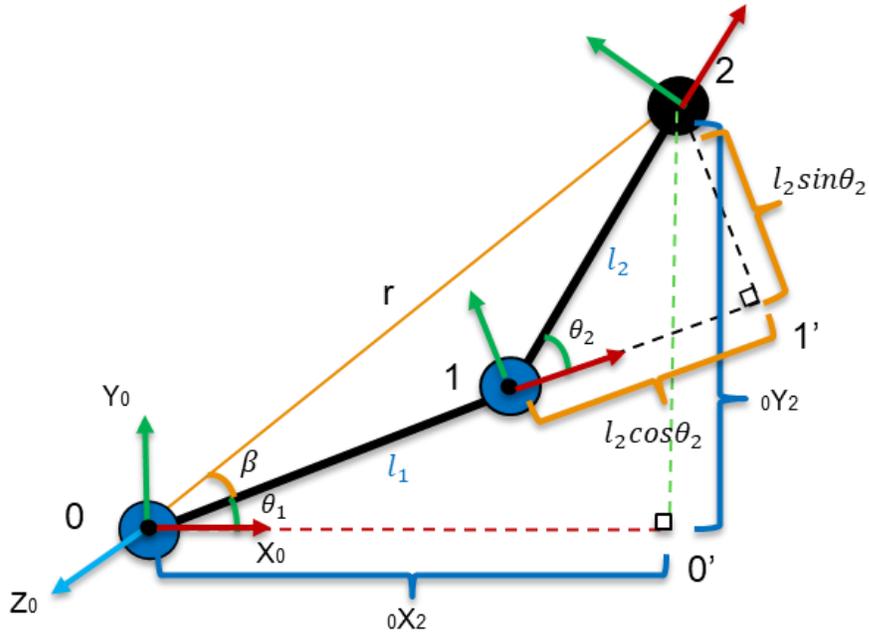


Figure 18: Trigonometric analysis for analytical IK

Consider $\Delta 021'$;

$$\beta = \operatorname{atan2} \left[\frac{l_2 \sin \theta_2}{l_1 + l_2 \cos \theta_2} \right]$$

Consider $\Delta O_2O'$;

$$\theta_1 = \operatorname{atan2} \left[\begin{matrix} {}^0Y \\ {}^2Y \\ {}^0X \\ {}^2X \end{matrix} \right] - \beta = \operatorname{atan2} \left[\begin{matrix} {}^0Y \\ {}^2Y \\ {}^0X \\ {}^2X \end{matrix} \right] - \operatorname{atan2} \left[\frac{l_2 \sin \theta_2}{l_1 + l_2 \cos \theta_2} \right]$$

Hence, we have now expressed the desired joint variables i.e. θ_1 and θ_2 in terms of known geometric parameters which include the link lengths and the end-effector positions. θ_1 will have two values corresponding to the elbow up and elbow down configurations as well. Here, we can choose θ_2 to be positive and calculate θ_1 which will give us values for the elbow down configuration.

CHAPTER 3

APPROACH AND RESULTS

The following objectives were planned:

- 1) Initial Testing: This objective enlists tasks which involve getting acquaintance with the developer tools, software and testing any procured components. Determining the coordinate frames of tracking and tracked components was also carried out as a part of this testing.
- 2) System Evaluation: Two main parameters had to be evaluated for establishing precision and accuracy of the system. Repeatability and cross-validation with an available system were the planned objectives. The tasks that were carried out for evaluation involve data acquisition techniques and data processing.
- 3) Quantification: Inverse kinematics is a process of obtaining joint variables from known spatial positions and orientations. Inverse kinematics was carried as a part of quantification. For this thesis work, it is based on a phantom prosthetic arm using the geometric or trigonometric approach. Additionally, since the system also outputs the orientations of the trackable object; using homogeneous transformation math and inverse orientation approach the yaw, pitch and roll angles between multiple trackers with respect to each other can also be determined which indicate the angles between the transformed components.
- 4) Custom Tracker Development: The objective was to develop an application specific tracker using the hardware development kit. The tasks driven by this objective include - CAD modeling, IMU and optical calibrations, sensor placement and raw data acquisition. A functional custom tracker was developed which would be optimized as a part of future work.

3.1 Initial Testing

3.1.1 COTS testing

The task aimed at testing the lighthouse technology using off-the-shelf components. A commercial VIVE tracker was used for tracking purposes. It was necessary to determine if the system worked satisfactorily in the absence of the virtual reality element that is the headset. The main idea driving this task was to get the IMU and optical sensor data from the system without the virtual reality element, i.e. the head-mounted display (HMD) and then replicate or reproduce similar data for the developed system i.e. the custom tracker. The custom tracker was important since these can be made specific to suit an application. The proposed application required a smaller form factor which would reduce any mass or inertia effects as well as it would mitigate any tissue artifacts when used for biomechanics motion capture. The geometry of the commercial VIVE tracker was also studied as well as the location of the sensors on the VIVE tracker. As previously indicated, the pose estimate relies on the position and number of photodiodes getting hit so, for the first prototype a similar placement was followed for the custom tracker. Additional details about the custom tracker have been documented in later sections.

3.1.2 Software tools acquaintance

A few developer tools are available from Valve for testing and determining tracking effectiveness such as SteamVR Tracking HDK, lighthouse console, VR tracking calibration, etc. The tools were used for raw data acquisition of the stated IMU and optical sampling parameters, for firmware updates, to aid auto placement of sensors and also for carrying out calibration routines for the IMU and the optical sensors or photodiodes. These tools were also explored during the initial testing.

3.1.3 Determining coordinate frames for tracking

It was observed that the tracker coordinate system did not align with the one in the provided documentation during testing. Hence, it was necessary to determine the coordinate frames for the system. This would also aid in later tasks such as cross-validation. As indicated before, using the lighthouse system without the head mounted display makes the base station the global frame for tracking. Coordinate testing involved moving the tracker with respect to the base station and analyzing the corresponding values generated using SteamVR plugin in Unity. A grid was drawn, and trackers were placed at various reference points to record the changes in pose output values. Since, Unity was used to design the tracking environment and GUI, the pose output followed a left-handed coordinate system. However, it was possible to output the raw SteamVR plugin data which included the homogeneous transformation matrices from the base station to the tracker which followed a right-handed coordinate system. The coordinate frame for the tracker lies on the bottom plane while that for the base station is located at the centre. Figure 19 shows these coordinate frames:

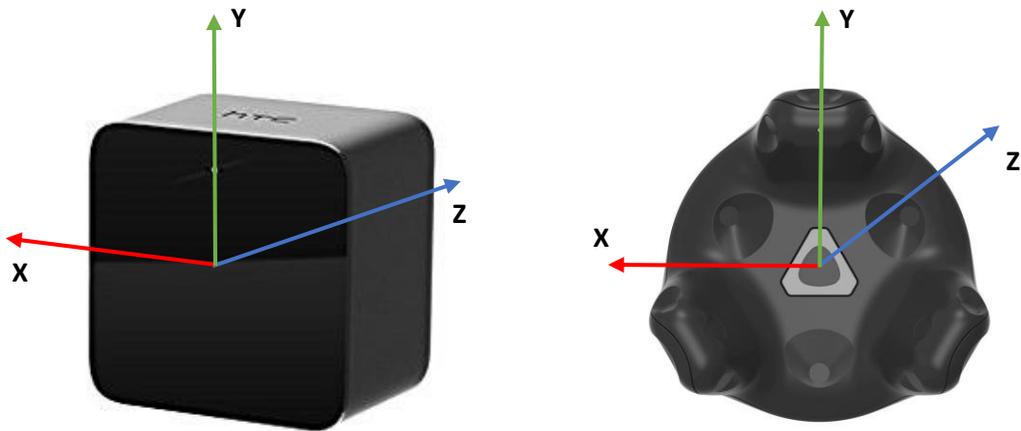


Figure 19: Base station and tracker coordinate systems

3.2 System Evaluation

3.2.1 Data acquisition

For evaluating the lighthouse technology, it was proposed to test the system functionality for tracking the end-effector of an industrial robot. The testing was carried out on a seven degree of freedom industrial robot arm at the Life Sciences Center at UMass Amherst. The purpose was to validate the system tracking against that of the robot arm to draw conclusions about the system repeatability and accuracy. A commercial VIVE tracker was mounted on the robot end-effector, and the position and orientation were obtained by the lighthouse technology using FTL's developed GUI in Unity. The position and orientation were also determined by using the robot's software. Three configurations were tested for ensuring consistency of data:

- 1) High: Base stations mounted high up w.r.t the robot and diagonally opposite to each other such that the master base station, the tracker, and the slave base station were approximately collinear.
- 2) Low: Base stations mounted at a low or equivalent height to that of the arm with the same collinear configuration as the High case.
- 3) High-Tri: Base stations mounted high up to form a triangle; the tracker and two base stations being the triangle vertices.

Figure 21 shows the position and orientation repeatability for 5 trials for one of the cases i.e. the high case. The following Table 2 summarizes the standard deviation results obtained using lighthouse technology for the three cases each with 5 trials where the data were sorted based on a common start frame since the robot arm motion was identical for all trials. Using the integrated software, the robot arm yielded sub-millimeter and sub-degree accuracy throughout. The gamma case for the robot arm data leads to a standard deviation error of about 3 mm as documented in Table 1 which might look like a discrepancy, however this is

due to the angle wrapping issue and not due to poor tracking or data loss. The angle wrapping issue also influences the standard deviation error for the lighthouse data. Hence, it should be noted that the actual orientation will not be as terrible as they seem from the table. A way of solving the angle flips has been discussed in subsequent sections. It was observed that the lighthouse system repeatability was not sub-millimeter or millimeter level as desired which would be a major hindrance for the precision applications that were proposed. However, further implications were drawn to aid future work based on observations during the testing. It should be noted that for convenience, the average standard deviation across all frames has been referred to as the standard deviation error.

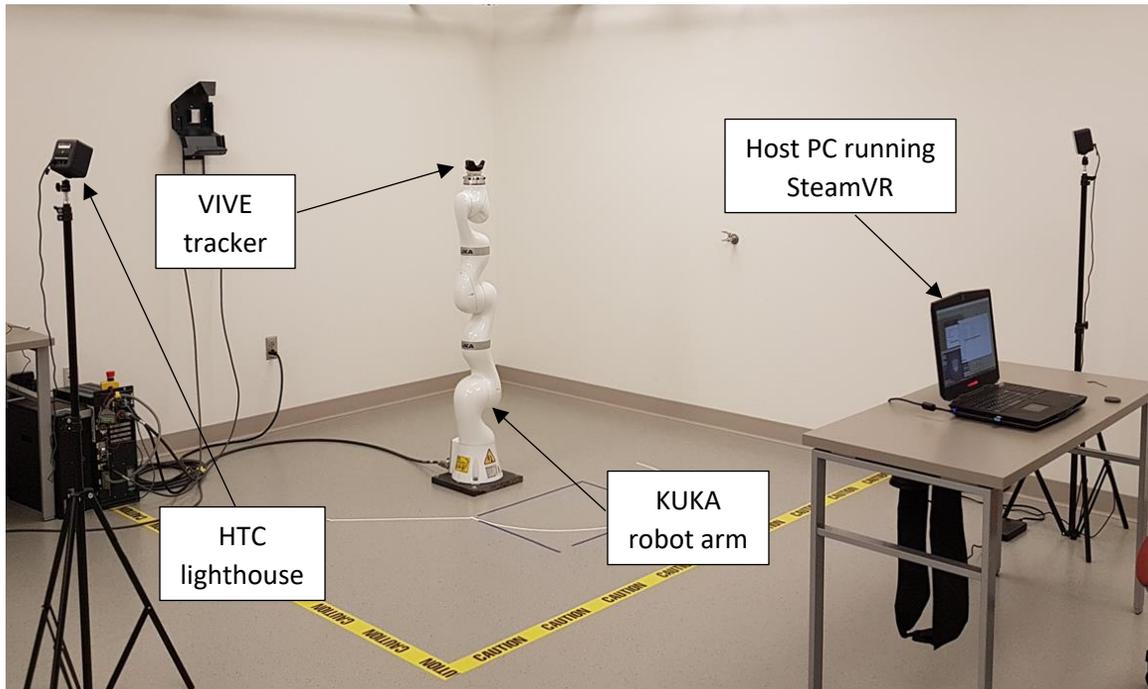


Figure 20: Lighthouse technology evaluation using a robotic arm

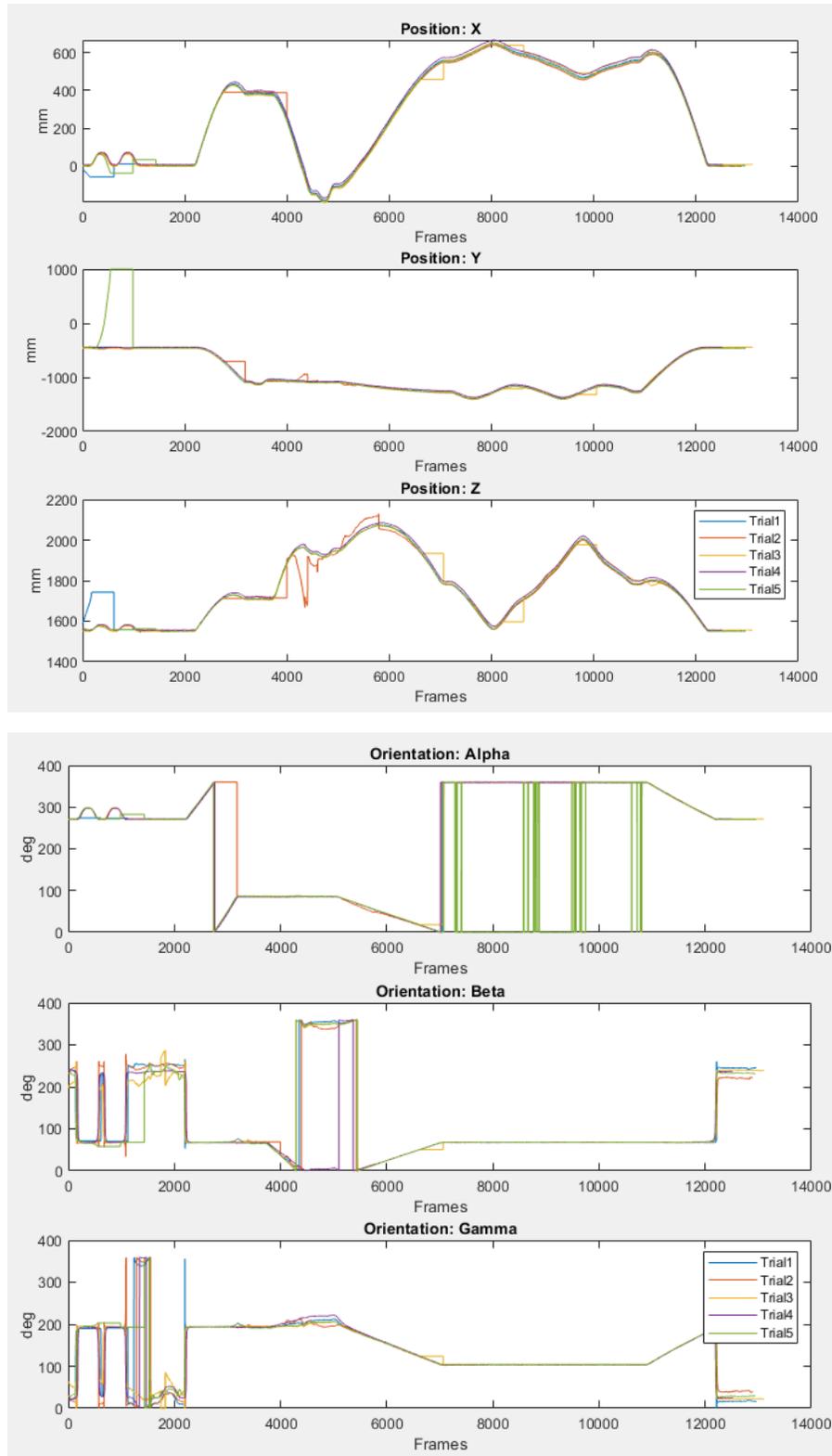


Figure 21: Repeatability plots for lighthouses mounted high up from initial data acquisition

Table 2: Repeatability results from the initial evaluation

Cases	X (mm)	Y (mm)	Z (mm)	alpha (deg)	beta (deg)	gamma (deg)
Robot arm High	0.6549	0.2937	0.6346	0.40107	0.10886	3.5809
LT Low	6.6	55.1	13.9	3.2682	19.8703	25.4256
LT High	13.1	44.5	14.8	36.1496	17.0666	8.9975
LT High-Tri	13.2	12.6	10.3	48.5062	6.2554	23.2709

For the evaluation using the robot arm, it was observed that the lighthouse system was losing considerable data characterized by the peaks in Figure 21. Similar issues were encountered in precision and accuracy estimation in reference [31]. During the initial evaluation, two lighthouses were used since the current system supports only a maximum of two base stations. However, the newer base stations which are currently being developed and which would be released in late 2018 would allow the use of more than two base stations which would take care of any occlusion related issues and data loss. Additionally, it was also noted that there was a significant offset in the trials which meant that even when the trials were aligned based on the frames, the global frame being used for recording the data was not constant. The lighthouse technology records its data in relation to the master base station and whenever there is any occlusion it keeps changing the master base station's global frame coordinates which leads to possible discrepancies. It was learnt from the forums that the SteamVR system updates the world level on every run and the positions and orientations are presented in a gravity-level Y up coordinate system for every update cycle [32]. These updates add inconsistencies in the data which would be referred to as global offset errors. The data obtained, can be refined using some post-processing methods or by using optimization routines. One such optimization routine employing the gradient descent algorithm was implemented by FTL to obtain relatively good results. Additional, data

acquisition methods were explored to record the data in order to avoid the global offset error by using a ground or reference tracker. This has been documented in subsequent sections.

3.2.2 Data processing for repeatability

Issues were encountered during data acquisition due to tracking loss and offset errors. This task involved post-processing or refining the data to get rid of these errors. The standard deviation or root mean square error minimization was the main objective driving this task. The entire data set was offset by a value such that the sets being compared superimposed as closely as possible thus driving the errors to minimum. A recursive loop that offsets the data set by a constant value and computes the overall error at each iteration was employed. The following procedure was implemented:

- i) The standard deviation error for all trials was calculated and the trial with the least standard deviation error was identified. This trial was treated as the reference trial for data refinement.
- ii) Consider Y to be the entity i.e. position or orientation which is a function of time or number of frames that needs to be refined. Let Y_i and Y_{ref} represent the corresponding functions or data sets to be refined, for the i^{th} trial and reference trial respectively. The objective is to minimize the norm of the error between the two data sets. The error is $(Y_i - Y_{ref})$ and the objective function can be written as reducing $||Y_i - Y_{ref}||$.
- iii) For the first iteration, the error and the error norm were computed.
- iv) A fixed step was taken in either directions about the i^{th} trial and if the error at the next iteration was found to be smaller than the error at previous iteration with respect to the reference trial, the procedure was repeated; while if the error increased in either directions towards about the i^{th} trial, then this would imply that the current

data set cannot be further refined. It was necessary to proceed in both directions with the initial step since the offset observed was positive in some cases as well as negative for other trials. A step in either direction would thus generalize the solution for trials with offsets in either directions about the reference trial.

- v) The termination criteria was the slope of the error. If the slope of the error at $(n+1)^{\text{th}}$ iteration was found to be positive or the error at $(n+1)^{\text{th}}$ iteration was more than the previous iteration, then the procedure was terminated.

Figure 22 shows one of the refined trials after implementing the above procedure.

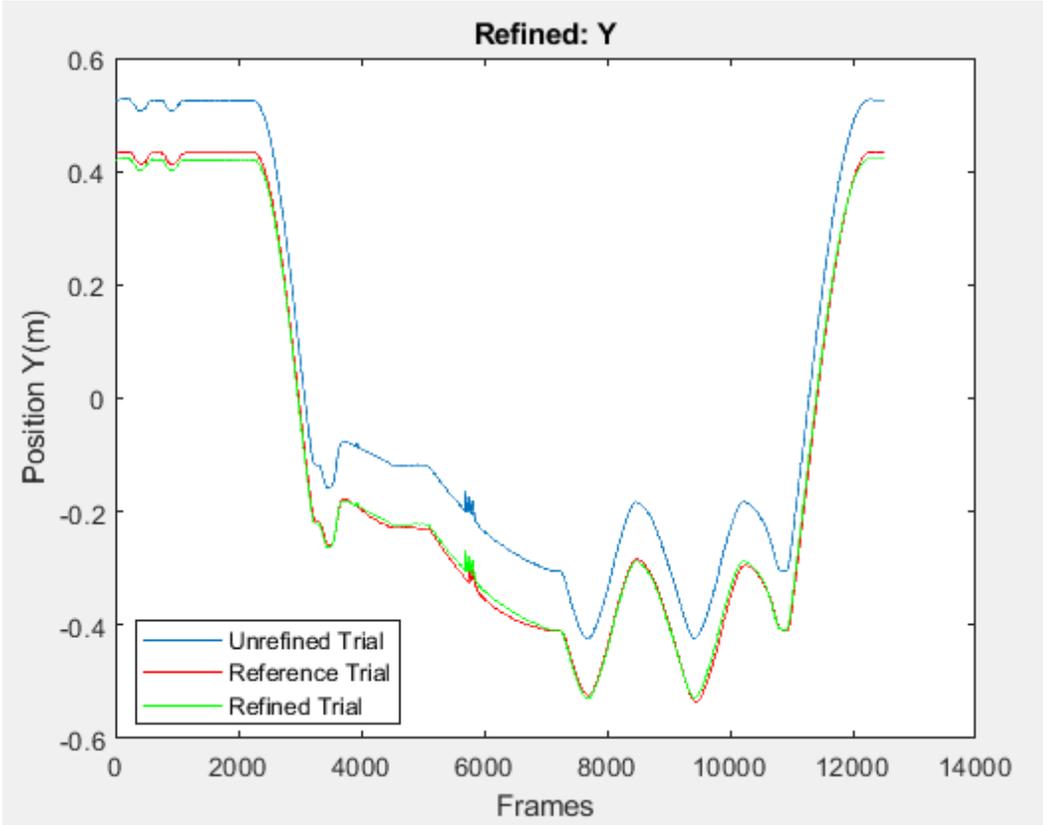


Figure 22: Refined Y trial post-processing for Low case

The following results were obtained for the High-Tri case after refining all the trials for the position data. Out of the 5 trials, one of the trials was ignored since there was a significant

amount of tracking loss observed. Figure 23 shows the refined data plots for the High-Tri case considering 4 trials.

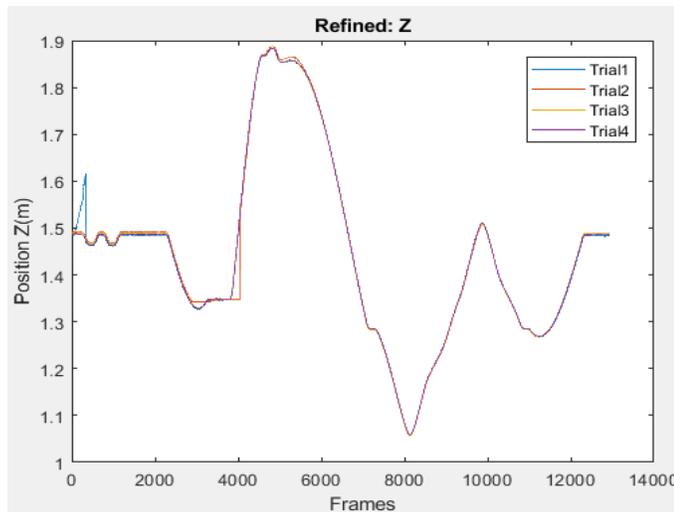
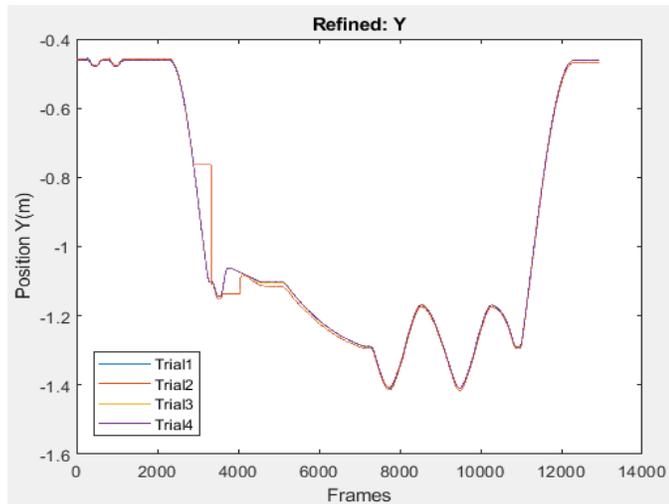
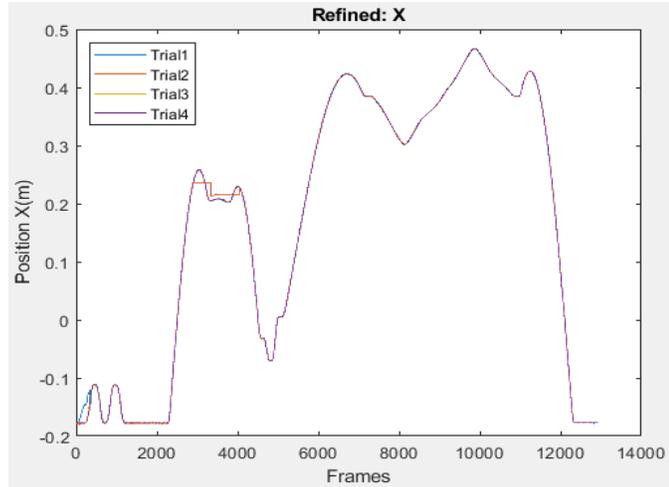


Figure 23: Refined position plots after data processing

The repeatability results obtained are documented in Table 3 based on standard deviation error amongst the 4 trials:

Table 3: Post-processing repeatability results for 4 trials

Position	Refined X (mm)	Refined Y (mm)	Refined Z (mm)
Standard Deviation Error	1.2	7.3	3.0

Additionally, if the three best trials were considered, where Trial 2 which had some tracking loss elements was discarded from the standard deviation set; the following results were obtained:

Table 4: Post-processing repeatability results for 3 trials

Position	Refined X (mm)	Refined Y (mm)	Refined Z (mm)
Standard Deviation Error	0.7548	0.7769	1.9414

The repeatability results obtained after post processing were sub and millimeter level repeatable for the position of the end-effector when tracking using the lighthouse technology. As for the orientation though, the results obtained after post processing did not look as good if the entire data set or all the frames were considered. For instance, consider the following gamma plot for the High-Tri case for 4 trials.

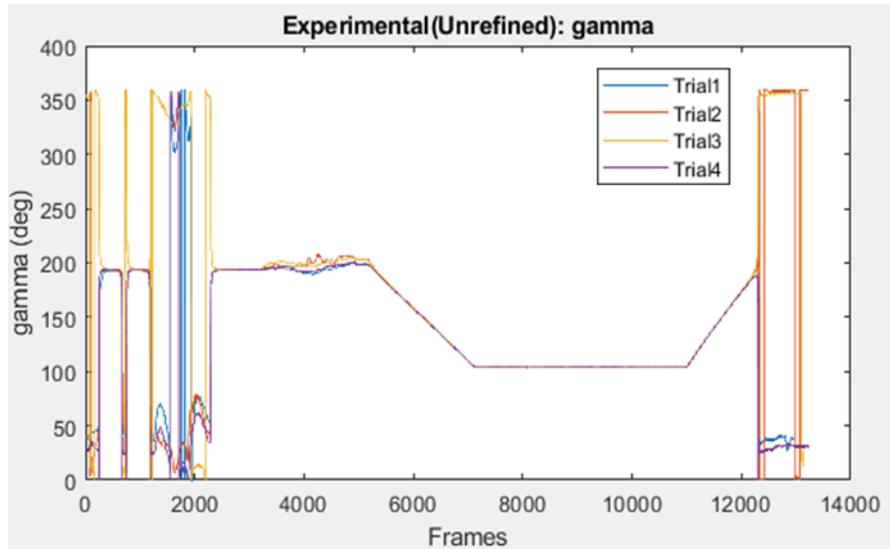


Figure 24: Gamma (Z rotation) plot for 4 trials

As seen in the Figure 24, the first 2500 frames and last 1000 frames have a lot of jumps characterized by the wrapping of the angles. This is because of any minor inaccuracies amongst two successive frames; causing a jump of 360 deg. For example, if the rotation value drops slightly below 0 deg say at the 6 or 7th decimal digit, still that would lead to a jump of 360 deg to indicate the start of a new rotation. Similar is the case for going over 360 deg. These are also common for 180 deg and -180 deg as well. A possible way to fix this issue is to add 360 deg to each successive frame where the difference between two consecutive frames exceeds a set threshold of say 360 deg. This is termed as unwrapping the angles. However, along with these snaps or jumps, significant data loss was observed for the stated intervals so unwrapping the angles did not do much of an improvement. However, if the region between frames 2500 and 12000 is considered which is about 75 % of the total interval; then sub degree level repeatability results can be achieved. It would be inappropriate for any practical application though to lose out or ignore any data.

The raw data obtained, or the experimental data obtained did not look good initially; hence it had to be post processed because of tracking loss and global offset errors. But, it would be

a hassle to post-process the data every single time. Additionally, any real time calculation would be delayed or affected by this. Hence, alternate data acquisition techniques had to be explored in order to improve the experimental data or the acquired raw data instead of post-processing it. The results until now, were obtained when the base stations were communicating wirelessly. Instead of having the base stations communicate wirelessly or based on optical syncing, a synchronization cable was used to see if the experimental data obtained was any better based on physical connection between the master and the slave base station. No significant improvements were observed based on this connection. One of the effective ways which showed promise was having a ground truth for tracking. This has been documented in the subsequent section.

3.2.3 Alternative data acquisition techniques

A possible solution to the global offset issue was to use a relative frame of reference for the SteamVR system. Since, the system updates the world frame at every run, it leads to corresponding offsets in the positions as well as orientations obtained from the trackers. These offsets are equivalent to the offset in the global frame caused by the frequent updates in the master base station coordinates. Hence, instead of using the base station as the global frame, a fixed reference point (an additional tracker fixed on the ground) would help eliminate this issue since the positions and orientations will be obtained with respect to this new frame of reference. The errors would persist in the transform from the base station to the ground frame, but they will not appear in the final transforms for the individual trackers. Though this would be laborious; since the positions and orientations will all have to be transformed w.r.t this fixed ground frame for each time frame every run, it is a possible work around for this issue.

Similar testing was carried out using the robot arm, with the same motion trajectory. A tracker was mounted on the end-effector of the KUKA LBR iiwa 14 R820. Additionally,

another tracker was set-up on the ground near the robot. The objective was to obtain the position and orientation of the end-effector tracker with respect to the ground tracker's coordinate frame. The principle here was that since an offset exists from the base station to the end-effector tracker; another offset would exist between the base station and the ground reference tracker and a transform that expresses the end-effectors coordinates in ground tracker's frame would nullify or balance out this offset. This transformation was done using homogeneous transformation math. The setup using a ground reference tracker has been shown in Figure:

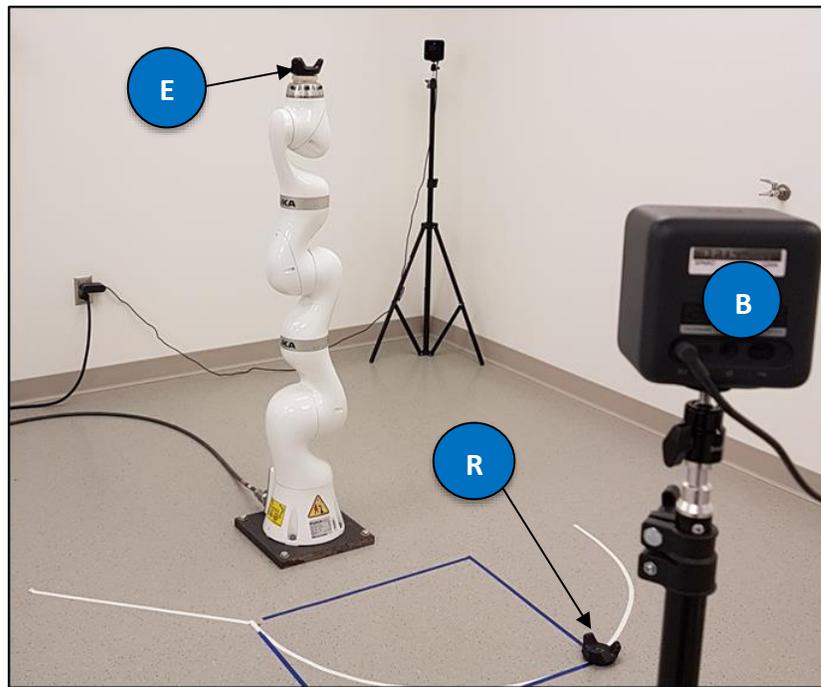


Figure 25: Setup for reference tracking using an additional ground tracker

Let E denote the end-effector tracker, R denote the ground reference tracker and B denote the base station coordinate frame. The experimental data during the testing expresses the positions and orientations of the end-effector and ground reference trackers in the base stations coordinate frame. It should be noted that it is possible to derive the 12 variable parameters consisting of the rotation matrices and position vectors of the homogeneous

transformation matrices for each individual tracker with respect to the base station by using the SteamVR plugin's raw data functionality. For our setup, we can represent the transformations again using a similar vector diagram as in Figure 26. Again, it should be noted that each of these components have their own coordinate systems oriented independent of one another.

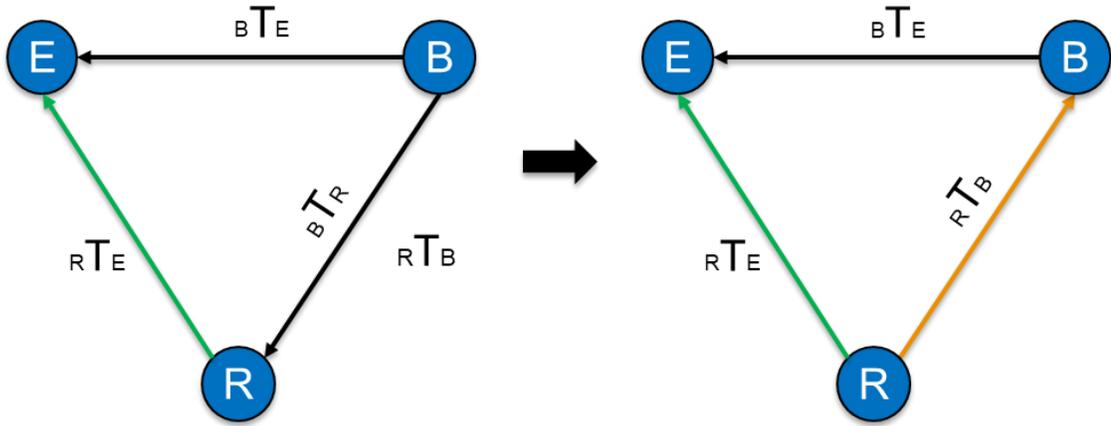


Figure 26: Ground reference tracking transformations using vector diagrams

The experimental data has been shown in black which includes the transforms T_E^B and T_R^B . The green vector shows the transform that we are interested in i.e. T_E^R . Based on transformation arithmetic, this can be written as:

$$T_E^R = (T_R^B)^{-1} \cdot (T_E^B) = (T_B^R) \cdot (T_E^B)$$

This process can be repeated for all frames. The positions x, y and z that express the point E (end-effector tracker) in the reference tracker's coordinate frame can then be decomposed from the homogeneous transformation matrix which include the elements: $T_E^R(1,4)$; $T_E^R(2,4)$; $T_E^R(3,4)$ of the 4x4 transformation matrix. The orientations though which are the yaw, pitch and roll values need to be calculated which are essentially the Euler angles. The procedure used gives the Euler angles from the rotation matrix. As it turns out, there are multiple solutions to this and one might not end up with a unique answer. However, following

a particular transformation sequence it is possible to calculate the angles. Here, the inverse i.e. ZYX sequence was followed also termed as Tait-Bryan angles [33].

The procedure was implemented to calculate the positions and orientations of the end-effector tracker with respect to the ground or reference tracker. Similar to previous testing, multiple trials were carried out to establish conclusions about the repeatability of the system. The repeatability results for 4 trials have been documented in Table 5.

Table 5: Repeatability results with ground reference tracking

Pose	X (mm)	Y (mm)	Z (mm)	alpha (deg)	beta (deg)	gamma (deg)
Standard deviation error	1.8841	1.7161	1.4230	0.0968	0.1092	0.2490

The following figures show these positions and orientations:

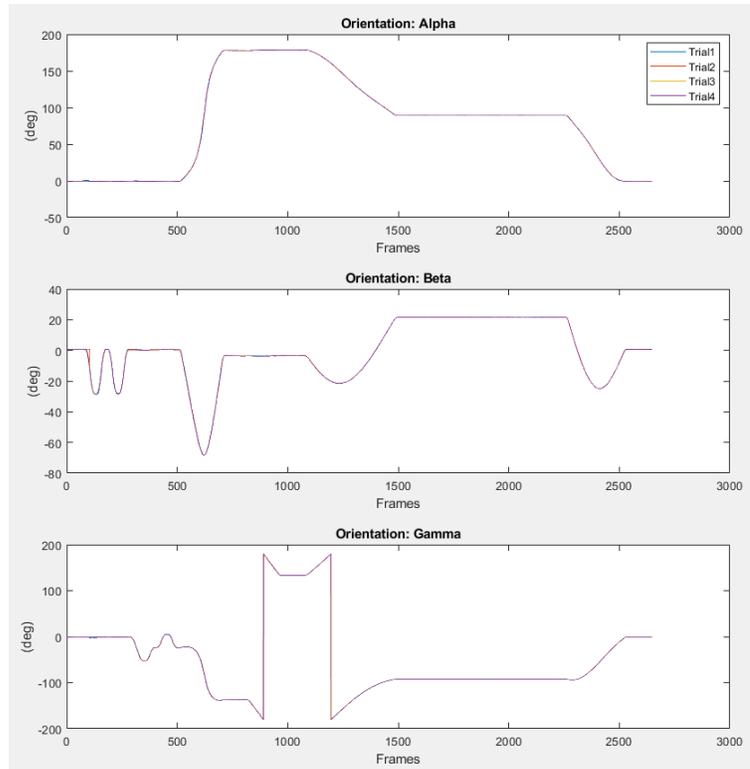


Figure 27: Repeatability orientation plots with ground reference tracking

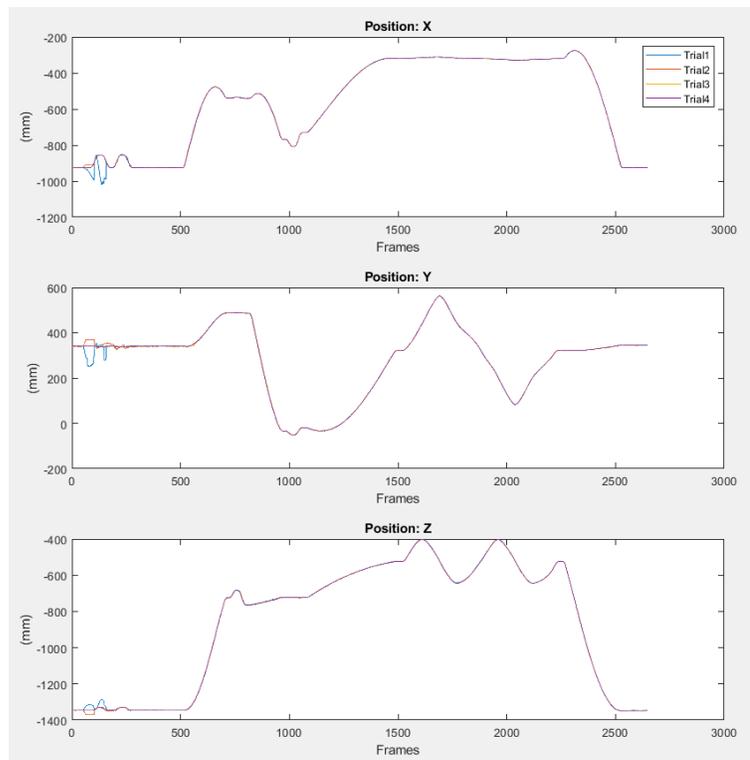


Figure 28: Repeatability position plots with ground reference tracking

As seen in Table 6; the results obtained are on a millimeter level for repeatability for position and sub-degree level for orientations. Additionally, as can be seen from the position data in Figure 28 there was some tracking loss in the beginning of the trials which can be attributed to less number of base stations or poor pose formation. If the starting frames from 0 to 175 are ignored; then considering 93.4 % of the trial; the following results are obtained based on repeatability:

Table 6: Repeatability results with ground reference tracking for 94 % interval

Pose	X (mm)	Y (mm)	Z (mm)	alpha (deg)	beta (deg)	gamma (deg)
Standard deviation error	0.6229	0.7808	0.8523	0.0928	0.08852	0.2545

These repeatability results show that it is possible to acquire precise data on a sub-millimeter and sub-degree level using the lighthouse system experimentally without any post processing.

3.2.4 Accuracy testing for the lighthouse system

Since the KUKA LBR iiwa 14 R820 and the lighthouse technology have different global frames for tracking, it is necessary to transform the obtained data from the robot arm's global frame to the lighthouse technology's global frame for cross-validation. For this task, however, the current validation method was to run a motion trajectory with the robot arm using the lighthouse technology and the camera-based capture system individually and then compare the results. Since the robot arm is highly repeatable on a sub-millimeter level, we can derive conclusions regarding system accuracy with this stated technique. A tracker was mounted on the end-effector and position and orientation data was acquired using the lighthouse

technology. Similarly, the position and orientation of the end-effector was obtained using Sunrise workbench, a software used for interfacing with the KUKA LBR iiwa 14 R820. Finally, the comparison of the two data sets was done for cross validation.

From repeatability testing, it was learnt that having a ground reference tracker improves the experimental data significantly without the need for post processing. Hence, all the further testing would involve using a ground reference tracker and having this tracker as the global frame for the lighthouse technology rather than the base station. The experimental setup for accuracy testing or cross-validation is the same as shown in Figure 29. However, we now have two global frames tracking the end-effector pose for each system.

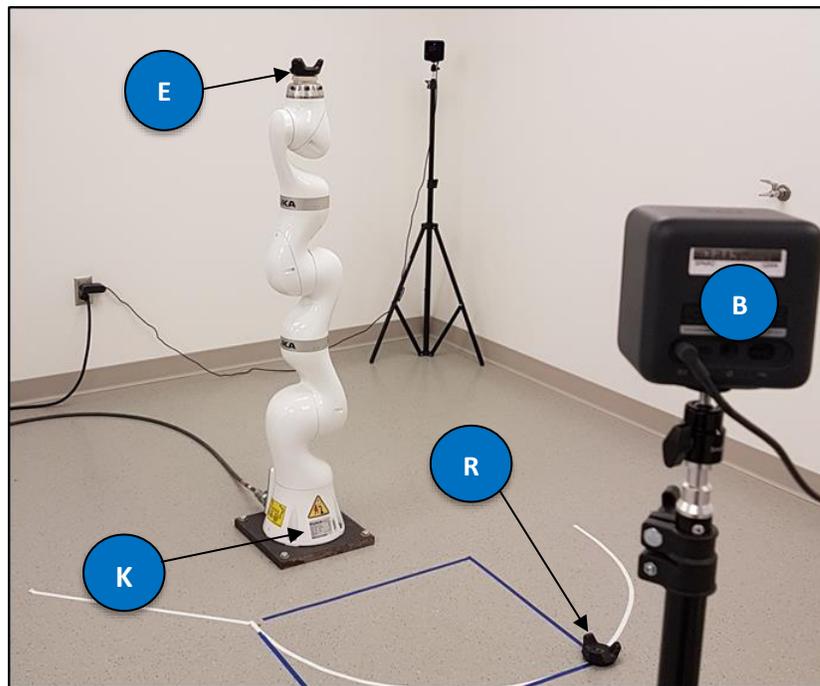


Figure 29: Cross-validation setup using KUKA LBR iiwa 14 R820

As discussed previously; B, E and R denote the base station, end-effector and ground reference tracker coordinate frames. K denotes the robot arm global frame as recorded by the software which is the centre point of the flange on which the robot is mounted.

3.2.4.1 Modeling the KUKA robot using forward kinematics

In order to determine the transform between the two global frames, the robot's end-effector pose had to be expressed as a transformation matrix at each frame. Since sunrise workbench can output the joint angles for the robot at each frame, using this functionality, the transformation matrices were calculated by modeling the robot arm. The forward kinematics model was constructed for the KUKA LBR iiwa 14 R820 using the modified D-H convention. The parameter Table 7 has been formulated based on the link lengths from the specification sheet for model LBR iiwa 14 R820 [34]. The mounting flange height was assumed to be 100 mm based on the experimental data during static trials. The link lengths have been assumed as per the specification sheet for the robot. The modified D-H table is as follows:

Table 7: Modified D-H table for KUKA LBR iiwa 14 R820

Link (i)	θ_i	α_{i-1}	a_{i-1}	d_i
1	θ_1	0	0	360
2	θ_2	$-\frac{\pi}{2}$	0	0
3	θ_3	$\frac{\pi}{2}$	0	420
4	θ_4	$\frac{\pi}{2}$	0	0
5	θ_5	$-\frac{\pi}{2}$	0	400
6	θ_6	$-\frac{\pi}{2}$	0	0
7	θ_7	$\frac{\pi}{2}$	0	126+100

It was decided to use the modified DH-table here, since it seems to be popular amongst relevant studies pertaining to industrial robots [35][36]. As compared to the standard DH table, the modified D-H table follows a different order of operations [37][38].

$$T_i^{i-1} = [\text{Rotn}(\alpha_{i-1})]_{x_{i-1}} \cdot [\text{Transn}(a_{i-1})]_{x_{i-1}} \cdot [\text{Rotn}(\theta_i)]_{z_i} \cdot [\text{Transn}(d_i)]_{z_i}$$

Hence, using the stated order of operations the transformation matrix from link (i-1) to link

(i) can be written as:

$$T_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i \cdot c\alpha_{i-1} & c\theta_i \cdot c\alpha_{i-1} & -s\alpha_{i-1} & -d_i \cdot s\alpha_{i-1} \\ s\theta_i \cdot s\alpha_{i-1} & c\theta_i \cdot s\alpha_{i-1} & c\alpha_{i-1} & d_i \cdot c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where, c and s are short for sine and cosine angles.

Based on this transformation matrix, and substituting the values for the known parameters,

we can write the transformation matrix for each link for the robot arm.

$$T_1^0 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & a_0 \\ s\theta_1 \cdot c\alpha_0 & c\theta_1 \cdot c\alpha_0 & -s\alpha_0 & -d_1 \cdot s\alpha_0 \\ s\theta_1 \cdot s\alpha_0 & c\theta_1 \cdot s\alpha_0 & c\alpha_0 & d_1 \cdot c\alpha_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 360 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & a_1 \\ s\theta_2 \cdot c\alpha_1 & c\theta_2 \cdot c\alpha_1 & -s\alpha_1 & -d_2 \cdot s\alpha_1 \\ s\theta_2 \cdot s\alpha_1 & c\theta_2 \cdot s\alpha_1 & c\alpha_1 & d_2 \cdot c\alpha_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^2 = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_2 \\ s\theta_3 \cdot c\alpha_2 & c\theta_3 \cdot c\alpha_2 & -s\alpha_2 & -d_3 \cdot s\alpha_2 \\ s\theta_3 \cdot s\alpha_2 & c\theta_3 \cdot s\alpha_2 & c\alpha_2 & d_3 \cdot c\alpha_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & 0 \\ 0 & 0 & -1 & -420 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^3 = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & a_3 \\ s\theta_4 \cdot c\alpha_3 & c\theta_4 \cdot c\alpha_3 & -s\alpha_3 & -d_4 \cdot s\alpha_3 \\ s\theta_4 \cdot s\alpha_3 & c\theta_4 \cdot s\alpha_3 & c\alpha_3 & d_4 \cdot c\alpha_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_4 & c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_5^4 = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & a_4 \\ s\theta_5 \cdot c\alpha_4 & c\theta_5 \cdot c\alpha_4 & -s\alpha_4 & -d_5 \cdot s\alpha_4 \\ s\theta_5 \cdot s\alpha_4 & c\theta_5 \cdot s\alpha_4 & c\alpha_4 & d_5 \cdot c\alpha_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 400 \\ -s\theta_5 & -c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_6^5 = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & a_5 \\ s\theta_6 \cdot c\alpha_5 & c\theta_6 \cdot c\alpha_5 & -s\alpha_5 & -d_6 \cdot s\alpha_5 \\ s\theta_6 \cdot s\alpha_5 & c\theta_6 \cdot s\alpha_5 & c\alpha_5 & d_6 \cdot c\alpha_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_7^6 = \begin{bmatrix} c\theta_7 & -s\theta_7 & 0 & a_6 \\ s\theta_7 \cdot c\alpha_6 & c\theta_7 \cdot c\alpha_6 & -s\alpha_6 & -d_7 \cdot s\alpha_6 \\ s\theta_7 \cdot s\alpha_6 & c\theta_7 \cdot s\alpha_6 & c\alpha_6 & d_7 \cdot c\alpha_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_7 & -s\theta_7 & 0 & 0 \\ 0 & 0 & -1 & -226 \\ s\theta_7 & c\theta_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transform from the robot's global frame to the end-effector is given by the product of all the above individual transforms:

$$T_7^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \cdot T_4^3 \cdot T_5^4 \cdot T_6^5 \cdot T_7^6$$

$$T_7^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where,

$$r_{11} = s7(s5(c4(s1.s3 - c1.c2.c3) - c1.s2.s4) - c5(c3.s1 + c1.c2.s3)) - c7(s6(s4(s1.s3 - c1.c2.c3) + c1.c4.s2) + c6.(c5(c4(s1.s3 - c1.c2.c3) - c1.s2.s4) + s5(c3.s1 + c1.c2.s3)))$$

$$r_{12} = s7(s6(s4(s1.s3 - c1.c2.c3) + c1.c4.s2) + c6(c5(c4(s1.s3 - c1.c2.c3) - c1.s2.s4) + s5.(c3.s1 + c1.c2.s3))) + c7.(s5.(c4.(s1.s3 - c1.c2.c3) - c1.s2.s4) - c5.(c3.s1 + c1.c2.s3))$$

$$r_{13} = c6(s4(s1.s3 - c1.c2.c3) + c1.c4.s2) - s6(c5(c4(s1.s3 - c1.c2.c3) - c1.s2.s4) + s5(c3.s1 + c1.c2.s3))$$

$$\begin{aligned}
r_{21} = & c7(s6(s4(c1.s3 + c2.c3.s1) - c4.s1.s2) + c6(c5(c4(c1.s3 + c2.c3.s1) \\
& + s1.s2.s4) + s5(c1.c3 - c2.s1.s3))) - s7(s5(c4(c1.s3 + c2.c3.s1) \\
& + s1.s2.s4) - c5(c1.c3 - c2.s1.s3))
\end{aligned}$$

$$\begin{aligned}
r_{22} = & -c7(s5(c4(c1.s3 + c2.c3.s1) + s1.s2.s4) - c5(c1.c3 - c2.s1.s3)) \\
& - s7(s6(s4(c1.s3 + c2.c3.s1) - c4.s1.s2) + c6(c5(c4(c1.s3 \\
& + c2.c3.s1) + s1.s2.s4) + s5(c1.c3 - c2.s1.s3)))
\end{aligned}$$

$$\begin{aligned}
r_{23} = & s6(c5(c4(c1.s3 + c2.c3.s1) + s1.s2.s4) + s5(c1.c3 - c2.s1.s3)) \\
& - c6(s4(c1.s3 + c2.c3.s1) - c4.s1.s2)
\end{aligned}$$

$$\begin{aligned}
r_{31} = & c7(c6(c5(c2.s4 - c3.c4.s2) + s2.s3.s5) - s6(c2.c4 + c3.s2.s4)) \\
& - s7(s5(c2.s4 - c3.c4.s2) - c5.s2.s3)
\end{aligned}$$

$$\begin{aligned}
r_{32} = & -c7(s5(c2.s4 - c3.c4.s2) - c5.s2.s3) - s7(c6(c5(c2.s4 - c3.c4.s2) \\
& + s2.s3.s5) - s6(c2.c4 + c3.s2.s4))
\end{aligned}$$

$$r_{33} = s6(c5(c2.s4 - c3.c4.s2) + s2.s3.s5) + c6(c2.c4 + c3.s2.s4)$$

$$\begin{aligned}
p_x = & 420.c1.s2 + 226.c6(s4.(s1.s3 - c1.c2.c3) + c1.c4.s2) - 226.s6(c5.(c4.(s1.s3 \\
& - c1.c2.c3) - c1.s2.s4) + s5(c3.s1 + c1.c2.s3)) + 400.s4(s1.s3 \\
& - c1.c2.c3) + 400.c1.c4.s2
\end{aligned}$$

$$\begin{aligned}
p_y = & 420.s1.s2 - 226.c6(s4.(c1.s3 + c2.c3.s1) - c4.s1.s2) + 226.s6(c5(c4(c1.s3 \\
& + c2.c3.s1) + s1.s2.s4) + s5.(c1.c3 - c2.s1.s3)) - 400.s4(c1.s3 \\
& + c2.c3.s1) + 400.c4.s1.s2
\end{aligned}$$

$$p_z = 420.c2 + 400.c2.c4 + 226.s6(c5.(c2.s4 - c3.c4.s2) + s2.s3.s5) + 226.c6(c2.c4 + c3.s2.s4) + 400.c3.s2.s4 + 360$$

$c1...c7$ and $s1...s7$ are short for cosines and sines for corresponding thetas. p_x, p_y, p_z represent the positions along X, Y and Z axes respectively for the end-effector; while $r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}$ form up the rotation matrix that represent the rotations for the end-effector frame with respect to the global frame. Respective Euler angles were calculated from this matrix which indicate the yaw, pitch and roll values between the frames. [32]. The values of joint angles were substituted in the above equations to obtain the position and orientations at each frame for the robot arm.

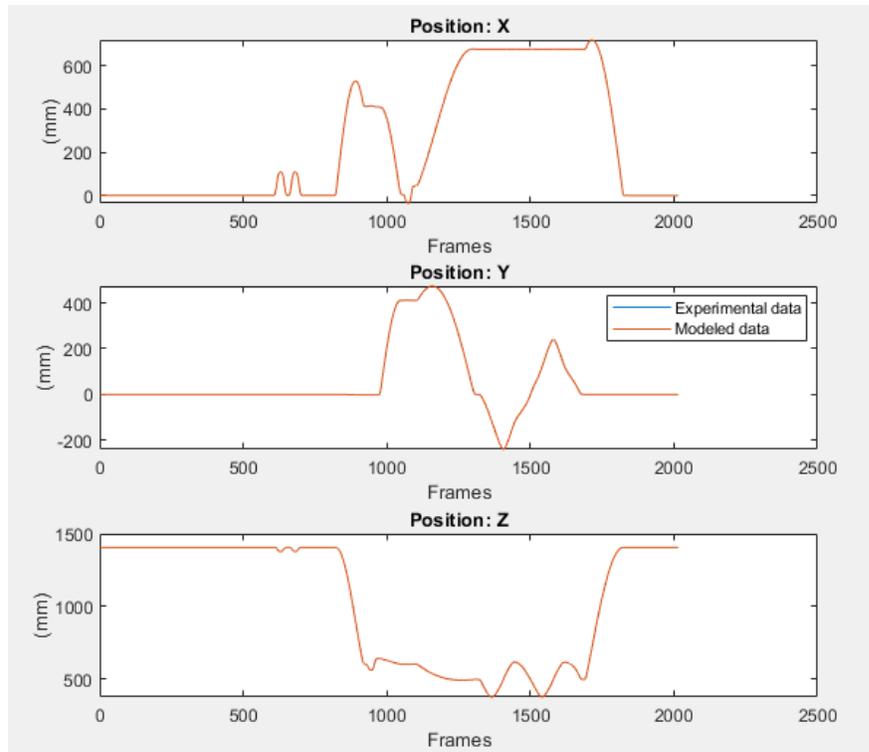


Figure 30: Position plots for experimental data and forward kinematics model

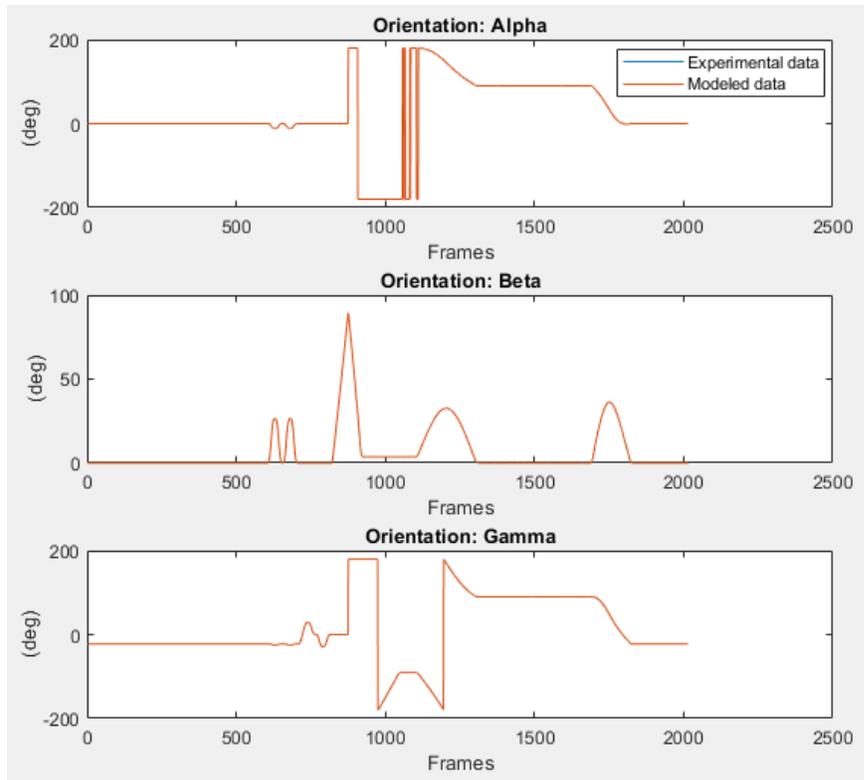


Figure 31: Orientation plots for experimental data and forward kinematics model

Table 8 shows the root mean squared error between the modeled data generated using forward kinematics and the experimental data.

Table 8: RMS errors between model and experimental data

Pose	X (mm)	Y (mm)	Z (mm)	alpha (deg)	beta (deg)	gamma (deg)
RMS error	0.1097	0.0720	0.1022	0.0097	0.0181	0.0189

Figure 30 and Figure 31 show the comparison between the collected experimental data and the model data generated using the above forward kinematics model for the KUKA robot.

3.2.4.2 Coordinate frame transformations for cross-validation

It was important for cross validation that the coordinate frames for the end-effector were aligned to establish a baseline for performing and comparing any transformations. From the coordinate testing done previously as a part of the initial testing; the trackers frame has been determined. As for determining the robot's global frame and the end-effector frame that Sunrise workbench uses to display the robot pose; a few static and dynamic trials were carried out. Consider that the robot is being viewed from the top view and side views, then Figure 32 shows the global frame orientation for the robot arm as well the coordinate frame for the end-effector as seen as a part of static trials.

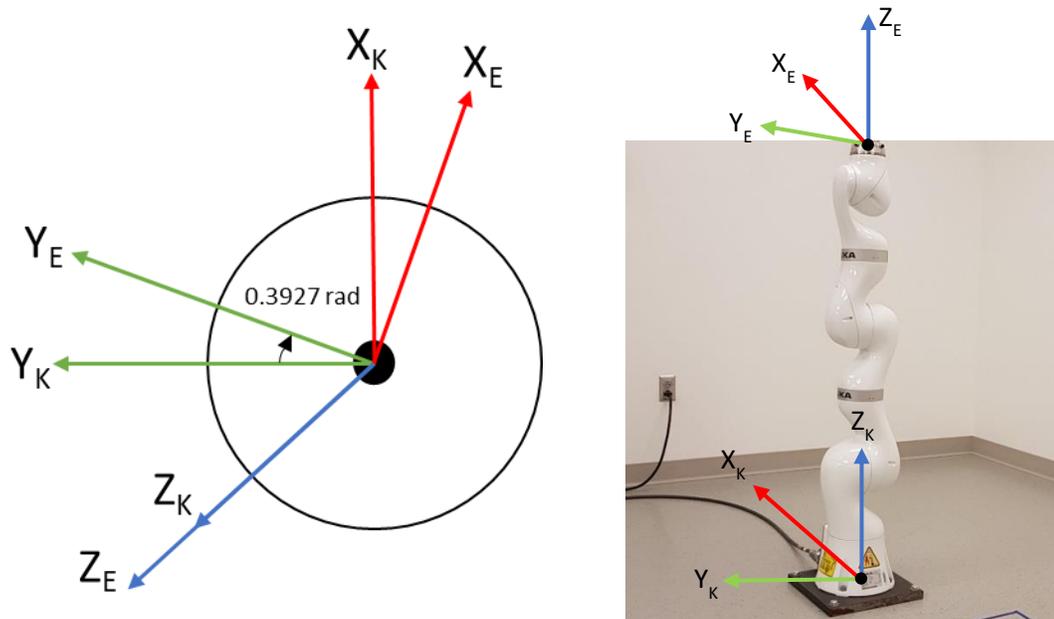


Figure 32: Coordinate frames for robot arm's global frame and end-effector

As indicated in Figure 32, it was found that for the home configuration or start position for all trials the last joint angle for the robot arm was -0.3927 rad which makes the gamma value -0.3927 rad in the obtained orientations for home configuration. Additionally, the Z value for the position of the end-effector was 1406 mm; while the robot is only 1306 mm in height and the mounting flange height was nowhere around 100 mm. This meant that the software was

assuming that a tool was mounted and indicated an offset point in space for the end-effector pose. The end-effector tracker's Y axis was aligned with the robot arm's global frame Y axis visually. The ground tracker was placed identical to the end-effector tracker. Figure 33 shows the tracker placement with their respective coordinate frames:

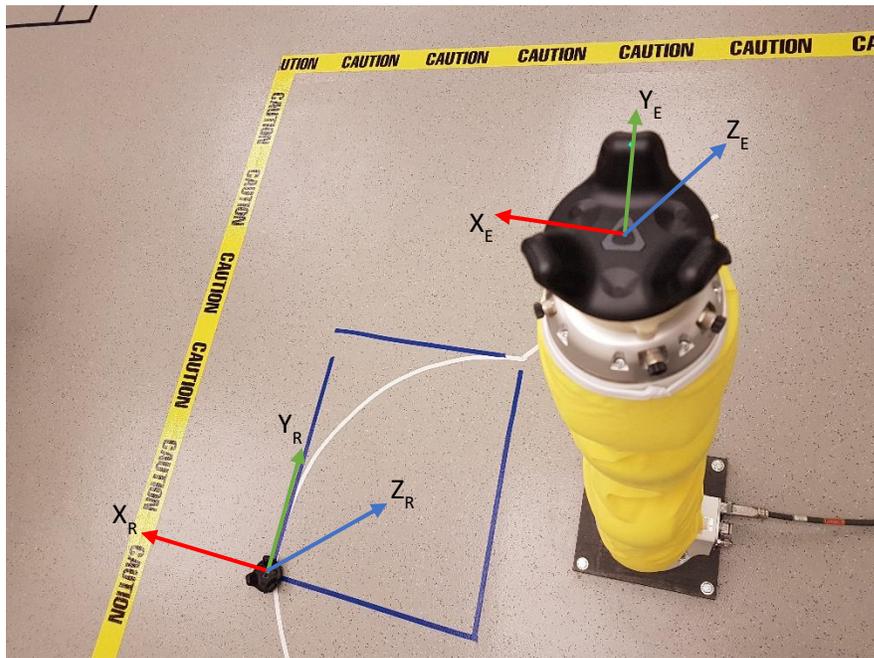


Figure 33: Tracker placement and their corresponding coordinate frames

From Figure 32 and Figure 33, it should be apparent that the end effector coordinate frames for the robot and that for the lighthouse technology have rotational as well as translational offsets between them. Since, there is no metric to do this, later it was discovered that there was an offset of about 2 deg which had to be accounted for in the transformations. It can also be noted that the KUKA robot is mounted on a base plate while the tracker ground reference tracker is on the floor. This induces an offset (translational) in the Z direction. It is important to get rid of all these offsets since they produce sequential offsets in the corresponding transformations which propagate to the final transform leading to eccentric results. These offsets can be solved in two ways, either by transforming the lighthouse end-effector tracker's coordinate frame to align with the robot arm's end effector coordinate frame or by

transforming the robot arm's end-effector frame to align with the lighthouse tracker's coordinate frame. Here, the later method was implemented. It was mentioned previously that the lighthouse end effector's Y axis matches up with the Y axis of the robot arm's global frame. Also, it is known that there is a rotation of -0.3927 rad (~22.5 deg) with a 2 deg offset about the Z axis between the KUKA robot's global and end effector frame. Hence, an inverse rotation of (0.3927 rad + offset) for the lighthouse end-effector tracker would align the Y axes for the two systems. KUKA LBR iiwa 14 R820 has a Z axis in the upward direction while the tracker has an upward negative Z axis. After aligning Y, we can fix the Z offset by rotating the aligned frame about the new X axis by 180 deg. The following two sequential transforms were performed to fix the rotational offsets:

$$\begin{aligned}
 \text{i)} \quad [Rotn(0.3927)]_{z_{i-1}} &= \begin{bmatrix} \cos(0.3927) & -\sin(0.3927) & 0 & 0 \\ \sin(0.3927) & \cos(0.3927) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 & \begin{bmatrix} 0.9239 & -0.3827 & 0 & 0 \\ 0.3827 & 0.9239 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 \text{ii)} \quad [Rotn(\pi)]_{x_i} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\pi) & -\sin(\pi) & 0 \\ 0 & \sin(\pi) & \cos(\pi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

Post-multiplying the obtained homogeneous transformation matrix (T_E^K) at each frame with the above sequential transformations thus solved the rotational offset issue. The translational offset was resolved by modeling the last joint length differently in the forward kinematics model. As documented in the last row of Table 7 for link 7, 100 mm was determined to be the mounting flange height for the model data to match up with experimental data based on static trials. The offset including the base plate height, the tracker mounting rings and the imaginary tool centre was determined to be 79 mm. Hence, the updated length for the last link was

considered as 147 mm. The new transformation matrix $(T_E^K)_{trans}$ represents the obtained transform after solving for these offsets.

Figure shows the KUKA LBR iiwa 14 R820 model data and the transformed coordinates after solving for the rotational and translational offsets.

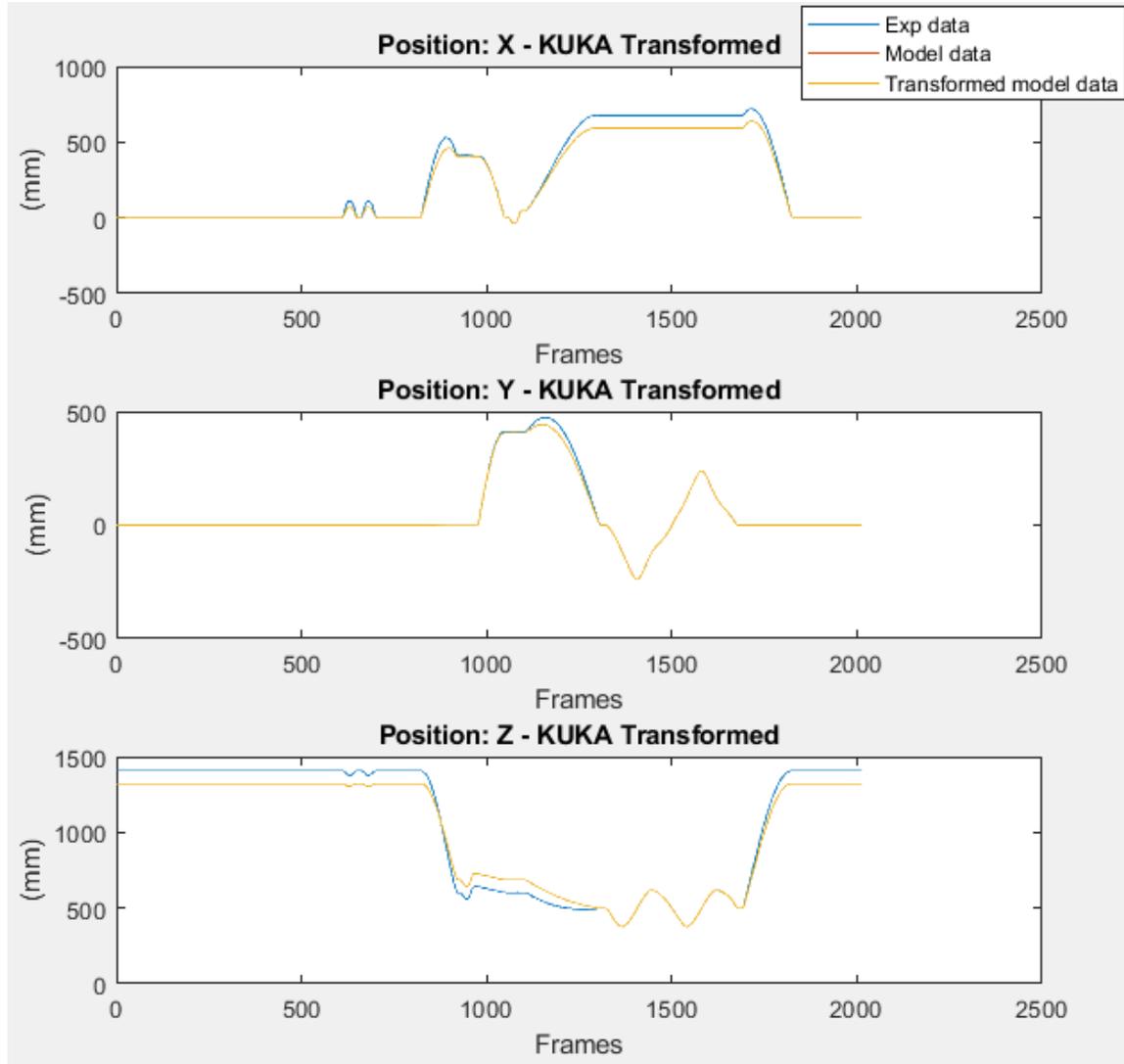


Figure 34: Transformed position data for KUKA robot

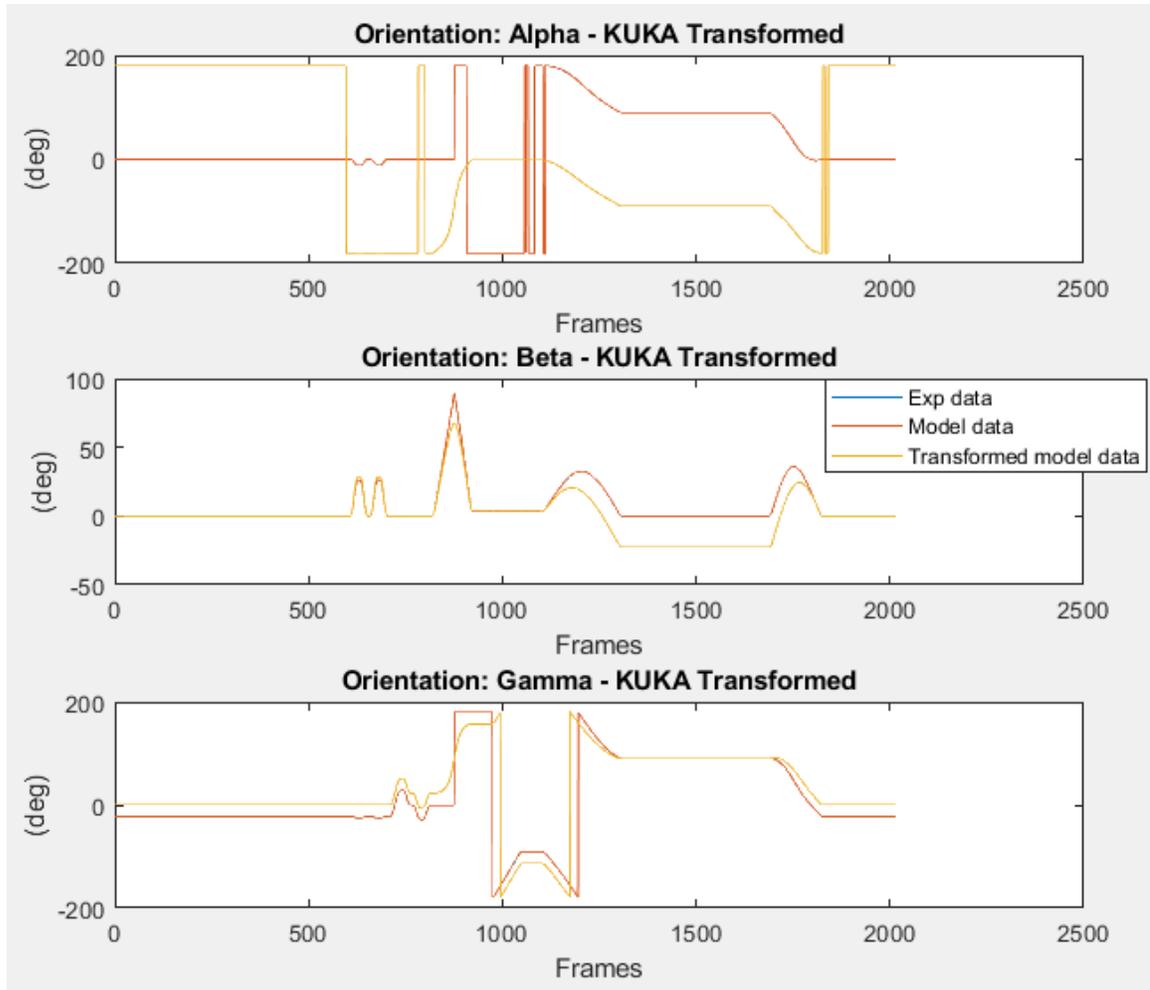


Figure 35: Transformed orientation data for KUKA robot

3.2.4.3 Global frame transform between the two systems

To determine the transforms between the global frames between the lighthouse technology and the robot arm; it was necessary that the same trials or rather the same time frames were being compared. However, it was noted that no matter if the end effector was moving or not, the transform between the two global frames should remain constant since the global frames are stationary. Hence, instead of calculating the transform from dynamic trial data, static trials were performed to do so. The position and orientation for the end-effector of the robot arm were recorded using Sunrise workbench and the lighthouse technology without moving the end-effector.

Consider the following transformation vector diagram as shown in Figure 36:

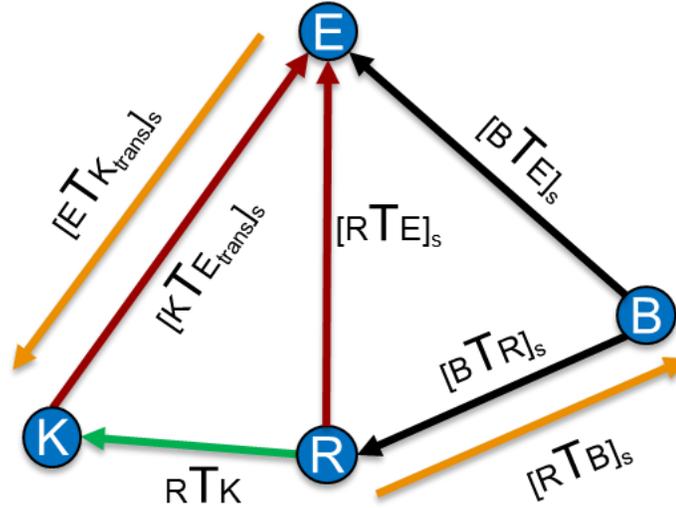


Figure 36: Transformation vector diagram for global frame transform using static trials

The black vectors in the diagram are the transforms that are available from the experimental data. B, R, K and E have been shown in Figure 29. The subscript s stands for static trials. As done previously, the global frame for tracking for the lighthouse system is set to be the ground tracker. Expressing end-effector tracker's frame in ground reference coordinates, we have:

$$(T_E^R)_s = (T_R^B)_s^{-1} \cdot (T_E^B)_s = (T_B^R)_s \cdot (T_E^B)_s$$

We are interested in the transform T_K^R as shown in green in the above vector diagram which represents the position and orientation of the KUKA robot's global frame in terms of the ground reference tracker frame. We now have $(T_E^R)_s$ from the static trials for the robot arm. $(T_E^K)_{trans_s}$ is the transformed matrix for the robot arm pose after performing the two sequential transforms for static trial transform T_E^K as stated in section 3.2.4.2 The global frame transform can be calculated as:

$$T_K^R = (T_E^R)_s \cdot (T_E^K)_{trans_s}^{-1} = (T_E^R)_s \cdot (T_K^E)_{trans_s}$$

From the static trials, it was found that the position and orientation readings for the lighthouse technology were consistent. One of the frames was used as the reference frame to get the following transform:

$$T_K^R = \begin{bmatrix} 0.9986 & -0.0516 & -0.0128 & -889.1561 \\ -0.0517 & -0.9987 & -0.0022 & 357.4105 \\ -0.0127 & 0.0029 & -0.9999 & -26.3639 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.2.4.4 Cross-validation between the two systems

In order to cross-validate the two systems, it was necessary to transform the coordinates of one system to the other system by using the transform between the global frames. We now consider the dynamic trials again meaning that the end-effector was subjected to the same motion trajectory as done before. The transformed robot arm pose matrix $(T_E^K)_{trans}$ for dynamic trials was pre-multiplied by the global frame transform T_K^R to transform the robot software data to lighthouse technology's global frame.

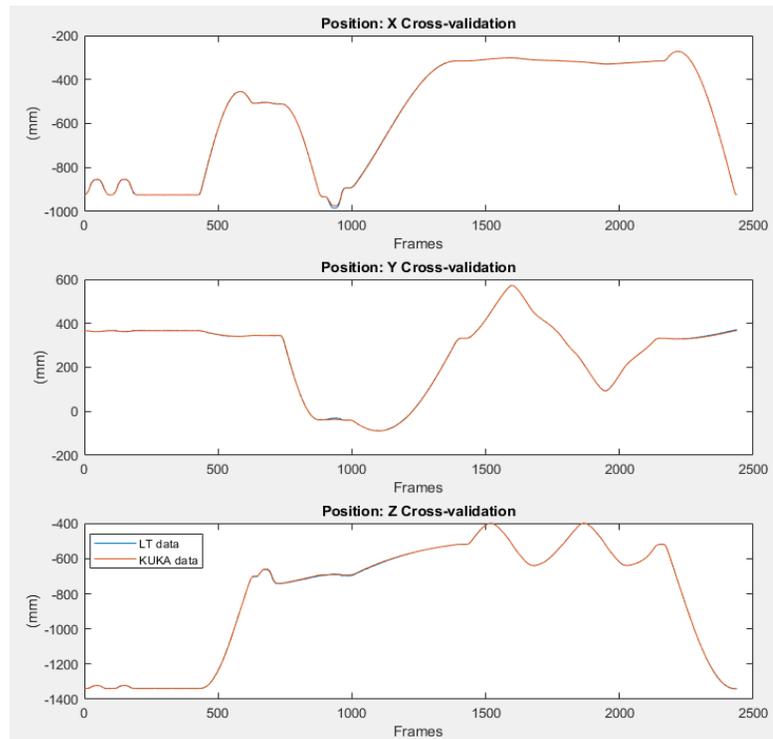


Figure 37: Position plots for cross-validation

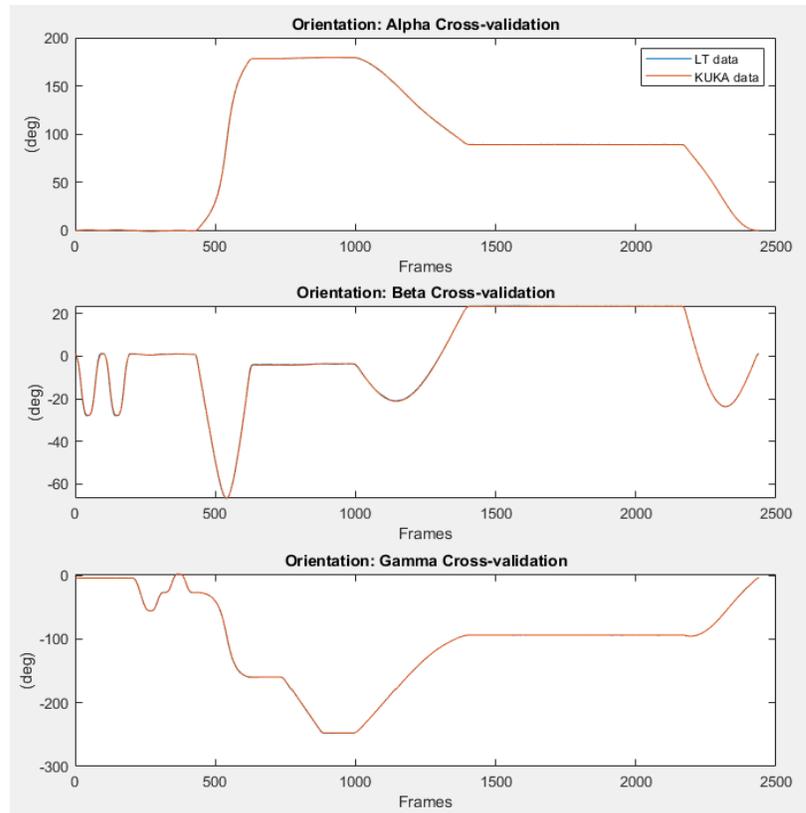


Figure 38: Orientation plots for cross-validation

Figure 37 and Figure 38 show the cross-validation results. It was observed that during the testing, for a few trials there was a slight time shift between the two data sets being compared which can be attributed to the lighthouse system. The raw data acquired from the system starts sampling at time zero and adds an increment which is the time period during the data acquisition for the system which is equivalent to the inverse of acquisition frequency for the system. The system software for the robot's pose, tracks the actual time though which is the hours of the day, minutes, seconds and milliseconds by using the UNIX time or Epoch time. The GUI for the lighthouse system starts sampling from 0 seconds. It was initially thought from the documentation that the system has a maximum update frequency of 60 Hz limited by the rotor rotational frequency for the laser sweeps. However, since the IMU samples at a much higher update rate of 250 Hz wirelessly, the optical data is augmented with the IMU data thus interpolating samples for the system at a higher update rate of more than 60 Hz.

For the above trials using FTL’s GUI an update rate of 100 Hz was possible. Using Fourier transforms it was possible to interpolate or up-sample the robot software data to match the lighthouse technology’s acquisition frequency. The lighthouse data was also down-sampled to compare with up-sampling. The following table summarizes the root mean squared errors between the two systems:

Table 9: RMS errors for system cross-validation

Pose	X (mm)	Y (mm)	Z (mm)	alpha (deg)	beta (deg)	gamma (deg)
RMS error (Up-sampling KUKA data)	1.6686	1.3570	1.7343	0.2272	0.2780	0.3266
RMS error (Down- sampling KUKA data)	1.6848	1.3555	1.7494	0.2269	0.2767	0.3249

As can be seen from Table 9, the accuracy of the system based on root mean squared error with respect to a high precision industrial robot arm is on a millimeter level scale for position and sub-degree level for orientation.

3.3 Quantification

3.3.1 Rigid body inverse kinematics approach [T4]

The key element that ties the captured motion data into useful information is inverse kinematics. As stated previously, the principle of quantifying motion using this technology is driven by the inverse kinematics approach where the determined positions and orientations

from the individual trackers would be used to calculate joint angles and relative translations between the links on which the trackers are mounted. Two ways of determining the joint angles have been carried out for rigid bodies. One uses multiple trackers and computes the homogeneous transformation matrices for each tracker in respective local frames to calculate the Euler angles by decomposing the homogeneous transformation matrix into rotation matrices, while the second method is based on a more analytical technique using a geometric or trigonometric approach that involves solving for the joint angles in terms of link lengths or known geometrical parameters when only the global positions and orientations of the last frame and the model frame are known. The parametric relations between the trackers' local frames and the global frame were determined using forward kinematics involving the Denavit - Hartenberg (D-H) convention for link labeling. A table summarizing the 4 D-H parameters was formulated which translates to the homogeneous transformations between each local frame established by the trackers. The forward kinematic relations between the links and joint angles aid in performing inverse kinematics as discussed in section 2.4

The following three configurations were tested using the phantom prosthetic arm.

- i) 1R (Single revolute joint)
- ii) 2R (Two revolute joints)
- iii) RRP (Two revolute joint and a prismatic joint)

3.1.1.1 1R configuration

Figure 39 shows the setup for performing inverse kinematics on the phantom limb.

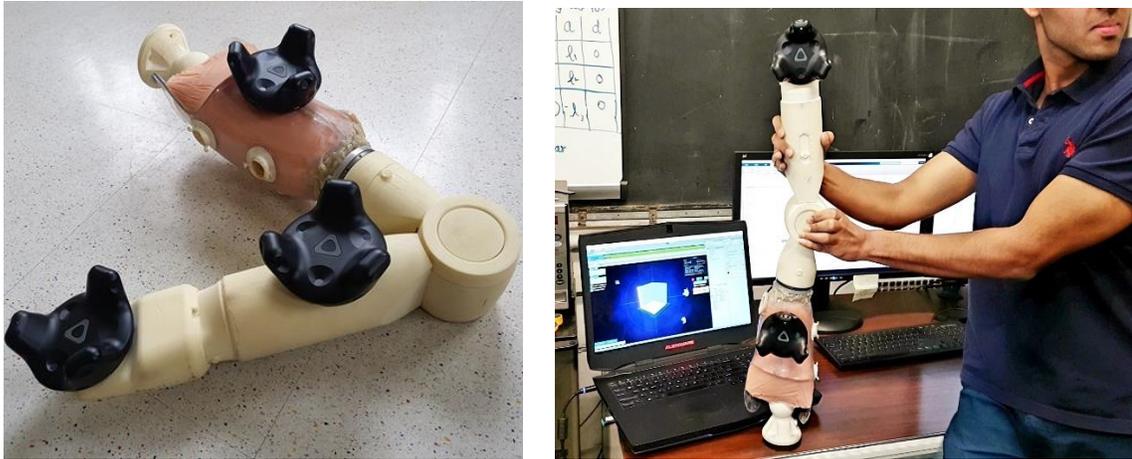


Figure 39: Setup for phantom limb inverse kinematics

For the 1R testing, consider the arrangement as shown in Figure 40.

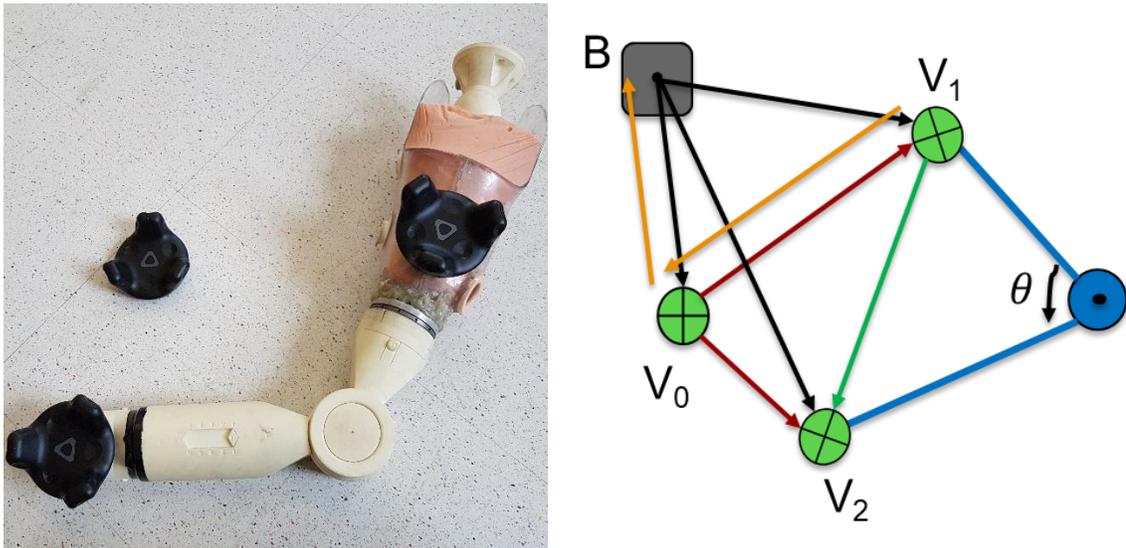


Figure 40: 1R Inverse kinematics testing using transformation arithmetic

We have three trackers here, where tracker V_0 is the ground tracker; while trackers V_1 and V_2 are mounted on the limb. The base station tracks all the trackers giving the transforms shown in black in the above diagram. Hence, from the experimental data we have $T_{V_0}^B$, $T_{V_1}^B$ and $T_{V_2}^B$. The objective is to calculate the elbow joint angle theta. Here, the trackers were so mounted that the Z axes of the three trackers and the axis of rotation of the joint were roughly parallel to each other. This approach relies on using transformation arithmetic to calculate

the rotation angle. By expressing the position and orientations of tracker V_2 in tracker V_1 frame, we know what the relative motion is between the two segments of the phantom limb.

The following transformations were performed:

Transform with respect to the ground reference frame:

$$T_{V_1}^{V_0} = (T_{V_0}^B)^{-1} \cdot T_{V_1}^B = T_B^{V_0} \cdot T_{V_1}^B$$

$$T_{V_2}^{V_0} = (T_{V_0}^B)^{-1} \cdot T_{V_2}^B = T_B^{V_0} \cdot T_{V_2}^B$$

Model tracker transform:

$$T_{V_2}^{V_1} = (T_{V_1}^{V_0})^{-1} \cdot T_{V_2}^{V_0} = T_{V_0}^{V_1} \cdot T_{V_2}^{V_0}$$

For the motion, three rotations were carried out in one direction; while the arm was subjected to another rotation in the opposite direction after the first three rotations. The quantification using the above approach further involved decomposing the homogeneous transformation matrices for the final transform from V_1 and V_2 into rotation matrices and finding out the respective Euler angles using the same approach as mentioned for previous tests. This can be also termed as the inverse orientation approach however; the name has been commonly associated with spherical wrists, but the principle is similar. As stated previously, the trackers were aligned to have their axes parallel. However, since this was done visually, there was an offset of about -8 deg initially. This offset can be resolved by using a rotational transform in the opposite direction with the same magnitude. Additionally, since the trackers were mounted on the surface it is obvious to note that there were a few translational offsets as well that had to be accounted for.

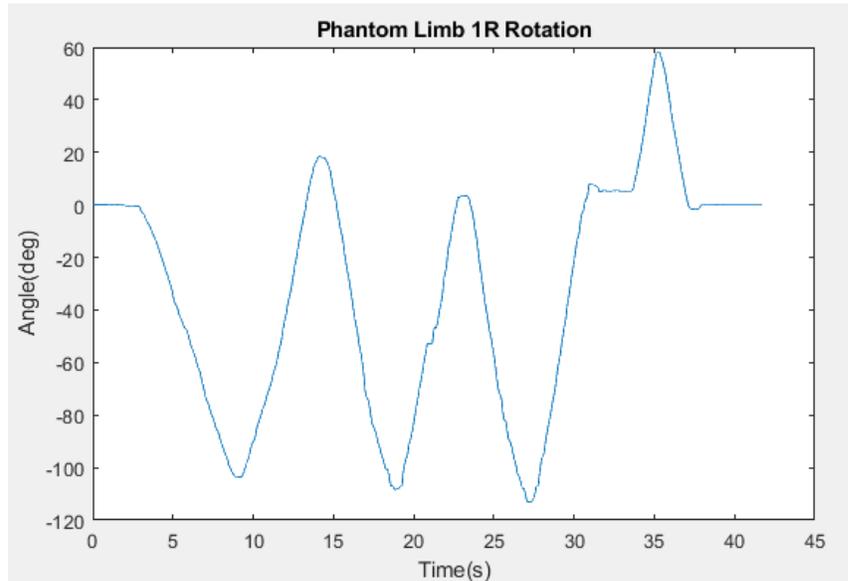


Figure 41: Elbow angle for the phantom limb

Figure 41 shows the elbow angle plot generated by using the stated approach.

3.3.1.2 2R configuration

The 2R testing or the setup involving two revolute joints was identical to the 1R testing setup since, inverse kinematics using analytical approach requires only the end-effector pose in the model or parent frame. As stated previously, there are several ways to perform inverse kinematics. However, since we have a system with less degrees of freedom, the analytical approach has been implemented here. For our setup, it is important to note that the two joints have their axes perpendicular to each other leading to a non-planar configuration. Forward kinematics was performed by assuming links and frames as per D-H convention. Consider the phantom arm in the home configuration. Figure 42 shows the frame layout for the system.

Forward Kinematics (2R):

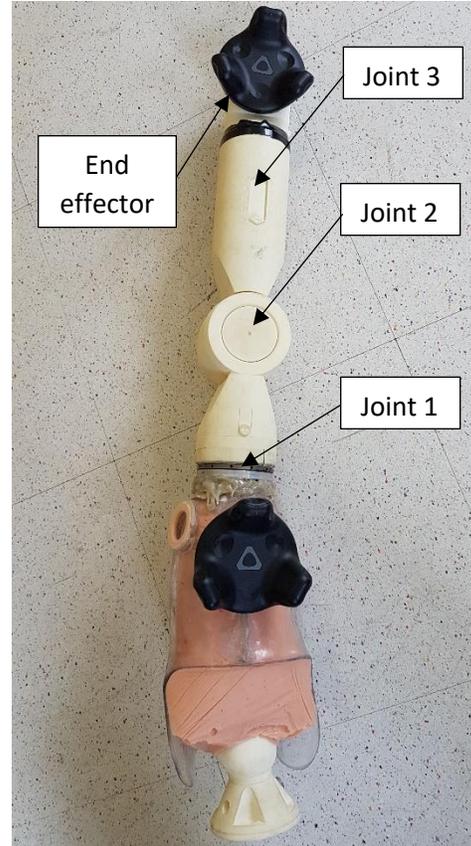
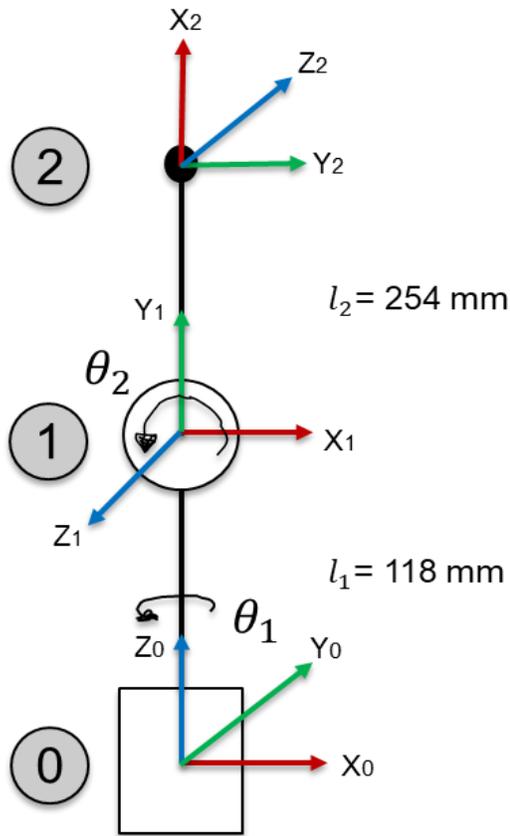


Figure 42: Link labeling and kinematic model of phantom limb

Table 10: D-H table for the phantom limb (2R case)

Link (i)	θ_i	α_i	a_i	d_i
	$[Z_{i-1}]$	$[X_i]$	$[X_i]$	$[Z_{i-1}]$
1	θ_1	$\frac{\pi}{2}$	0	118
2	$\theta_2 + \frac{\pi}{2}$	π	254	0

The transforms between the links are:

$$T_1^0 = \begin{bmatrix} c\theta_1 & 0 & s\theta_1 & 0 \\ s\theta_1 & 0 & -c\theta_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_1 & 0 & s\theta_1 & 0 \\ s\theta_1 & 0 & -c\theta_1 & 0 \\ 0 & 1 & 0 & 118 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} -s\theta_2 & c\theta_2 & 0 & -l_2 s\theta_2 \\ c\theta_2 & s\theta_2 & 0 & l_2 c\theta_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -s\theta_2 & c\theta_2 & 0 & -254 \cdot s\theta_2 \\ c\theta_2 & s\theta_2 & 0 & 254 \cdot c\theta_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The final transform can be written as:

$$T_2^0 = T_1^0 \cdot T_2^1 = \begin{bmatrix} -c\theta_1 \cdot s\theta_2 & c\theta_1 \cdot c\theta_2 & -s\theta_1 & -254 \cdot c\theta_1 \cdot s\theta_2 \\ -s\theta_1 \cdot s\theta_2 & c\theta_2 \cdot s\theta_1 & c\theta_1 & -254 \cdot s\theta_1 \cdot s\theta_2 \\ c\theta_2 & s\theta_2 & 0 & 254 \cdot c\theta_2 + 118 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The end-effector positions from forward kinematics equations are:

$$X_2^0 = -254 \cdot c\theta_1 \cdot s\theta_2$$

$$Y_2^0 = -254 \cdot s\theta_1 \cdot s\theta_2$$

$$Z_2^0 = 254 \cdot c\theta_2 + 118$$

From the relations for X_2^0 and Y_2^0 , we have;

$$\frac{Y_2^0}{X_2^0} = \frac{-254 \cdot s\theta_1 \cdot s\theta_2}{-254 \cdot c\theta_1 \cdot s\theta_2} = \tan(\theta_1)$$

$$\text{Hence; } \theta_1 = \tan^{-1}(Y_2^0, X_2^0)$$

From, the forward kinematic relations, we have already found a solution for one of the angles.

We now need to find the other joint angle: θ_2 .

Inverse Kinematics (2R):

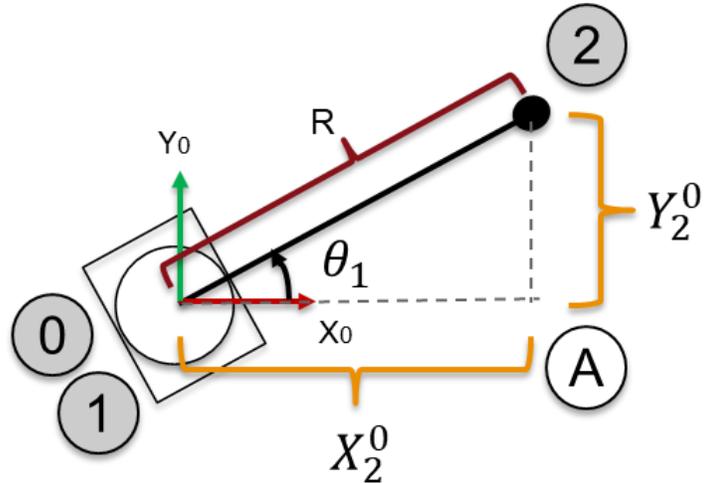


Figure 43: Top view analysis of the phantom arm

From Figure 43 considering $\Delta 02A$ or $\Delta 12A$, it can be deduced that:

$$\theta_1 = \tan^{-1}(Y_2^0, X_2^0)$$

The above relation confirms our derived equation from forward kinematics for θ_1 .

$$R = (X_2^0)^2 + (Y_2^0)^2$$

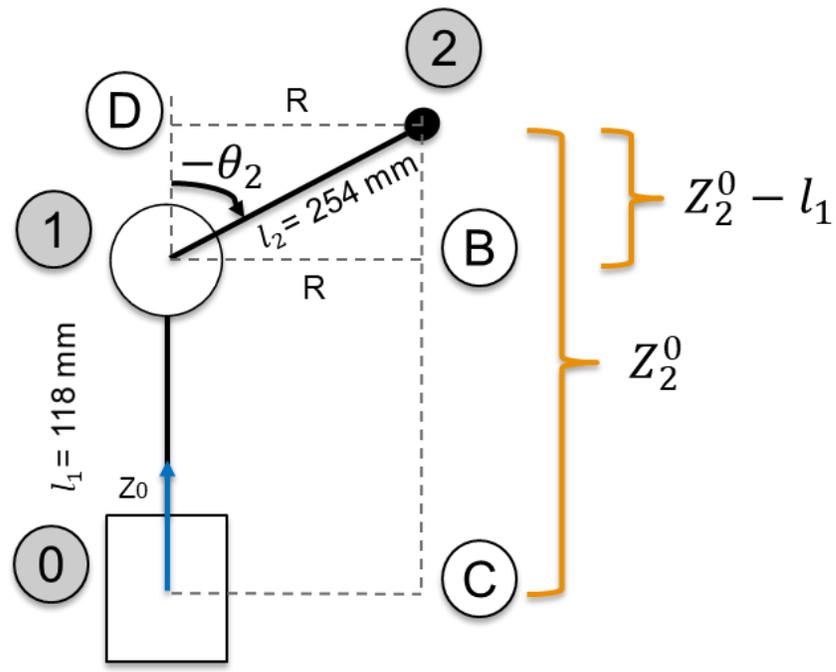


Figure 44: Side view analysis of the phantom arm

From the side view, as shown in Figure 44, it can further be noted that:

$$2B = Z_2^0 - l_1$$

Hence,

$$\theta_2 = -\tan^{-1}(R, Z_2^0 - l_1) = -\tan^{-1}(R, Z_2^0 - 118)$$

We know; $R = (X_2^0)^2 + (Y_2^0)^2$... *Pythagoras Thm.*;

$$\text{Hence, } \theta_2 = -\tan^{-1}(\sqrt{(X_2^0)^2 + (Y_2^0)^2}, Z_2^0 - 118)$$

Here, using the analytical approach we now have expressed θ_1 and θ_2 in terms of the end-effector pose and known joint parameters.

Figure 46 shows the two angles that were calculated using the above inverse kinematics approach for the phantom limb. The motion for the limb involved moving the arm along its first joint three times back and forth. After this motion, the second joint was subjected to three oscillations as well similar to the first joint. Figure 45 shows the start position during the testing. It can be seen that the angle θ_2 is slightly greater than 90 deg during the start of the trial, while θ_1 was about 30 deg initially. We see that these angles were quantified using analytical IK based on data using the lighthouse system.



Figure 45: Start position for 2R case inverse kinematics

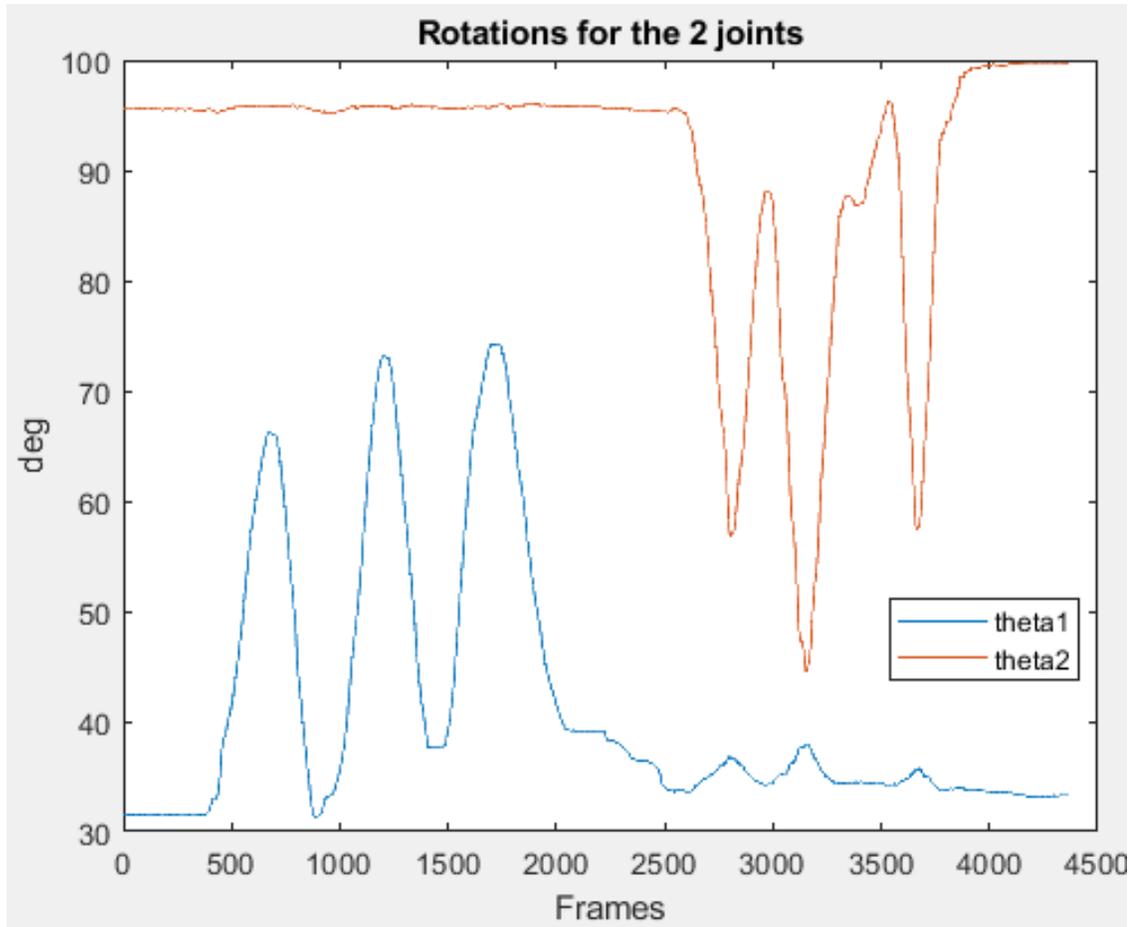


Figure 46: θ_1 and θ_2 for phantom arm (2R case)

3.3.1.3 RRP configuration

For the RRP configuration, the phantom arm was subjected to rotation along two revolute joints which were discussed as a part of the 2R testing. Additionally, since the phantom limb also has a translating element, the system can be compared to an RRP manipulator with 2 revolute and one prismatic joint. The link labelling was again carried out based on the D-H approach. We can again resort to forward kinematics which would help us obtain a possible relation for the joint variables from the final equations for the end-effector pose, however as seen previously, we were able to derive a closed form solution for the two angles using the trigonometric approach using inverse kinematics which was verified using the forward kinematics relation. Hence, for the RRP configuration, only the inverse kinematics approach

has been shown. As before we are interested in the joint angles, but additionally also in the translation of the end-effector.

Inverse Kinematics (RRP):

For the RRP configuration, an additional prismatic joint was considered as shown in Figure 47 by joint 2.

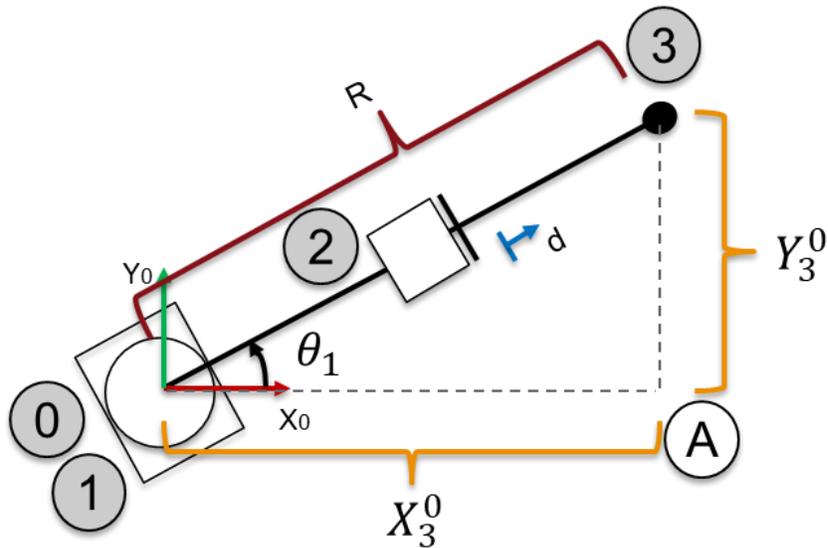


Figure 47: Top view analysis for the RRP configuration

From Figure 47 considering $\Delta 03A$ or $\Delta 13A$, it can be noted that:

$$\theta_1 = \tan^{-1}(Y_3^0, X_3^0)$$

It would seem from the figure that the length R is equivalent to: $(l_2 + l_3 + d)$. However, this is not true since the length 03 or 13 is the projection of the entire length we are considering on the XZ plane. This length would change depending on the angle made by the second joint since, it should be noted that the angle θ_2 might not be in zero configuration. The following equation can be written based on Pythagoras theorem for R :

$$R = (X_3^0)^2 + (Y_3^0)^2$$

Now, we consider the side view analysis as shown in Figure 48.

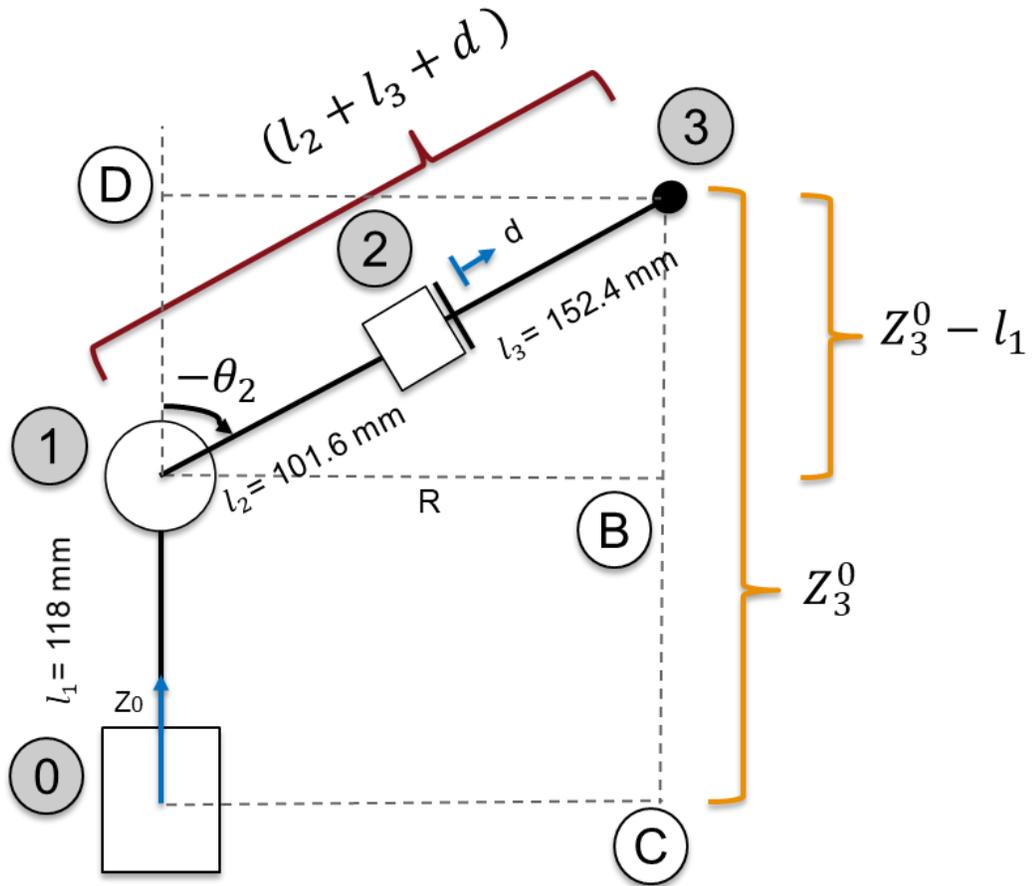


Figure 48: Side view analysis for the RRP configuration

Consider $\Delta 13B$,

$$\theta_2 = -\tan^{-1}(R, Z_3^0 - l_1) = \tan^{-1}(\sqrt{(X_3^0)^2 + (Y_3^0)^2}, Z_3^0 - l_1)$$

We now have found the two joint angles θ_1 and θ_2 . We still need to find the translation of the last joint i.e. d .

From $\Delta 13B$ it can be further noted by Pythagoras theorem that,

$$(l_2 + l_3 + d)^2 = (Z_3^0 - l_1)^2 + R^2$$

$$(254 + d)^2 = (Z_3^0 - 118)^2 + ((X_3^0)^2 + (Y_3^0)^2)$$

$$d = \sqrt{(Z_3^0 - 118)^2 + ((X_3^0)^2 + (Y_3^0)^2)} - 254$$

The above equations were implemented for the testing data obtained from the trackers for the phantom arm. There were offsets from the model and actual frames which again had to be resolved by means of individual transformations. The procedure is similar to studies by Corke [39]. The phantom arm was subjected to similar motion like the 2R case i.e. three oscillations along the first two joint. Along with rotations, the last joint was translated as well during the motion. Figure 49 and Figure 50 show the angles and translations for the motion based on the above inverse kinematics approach:

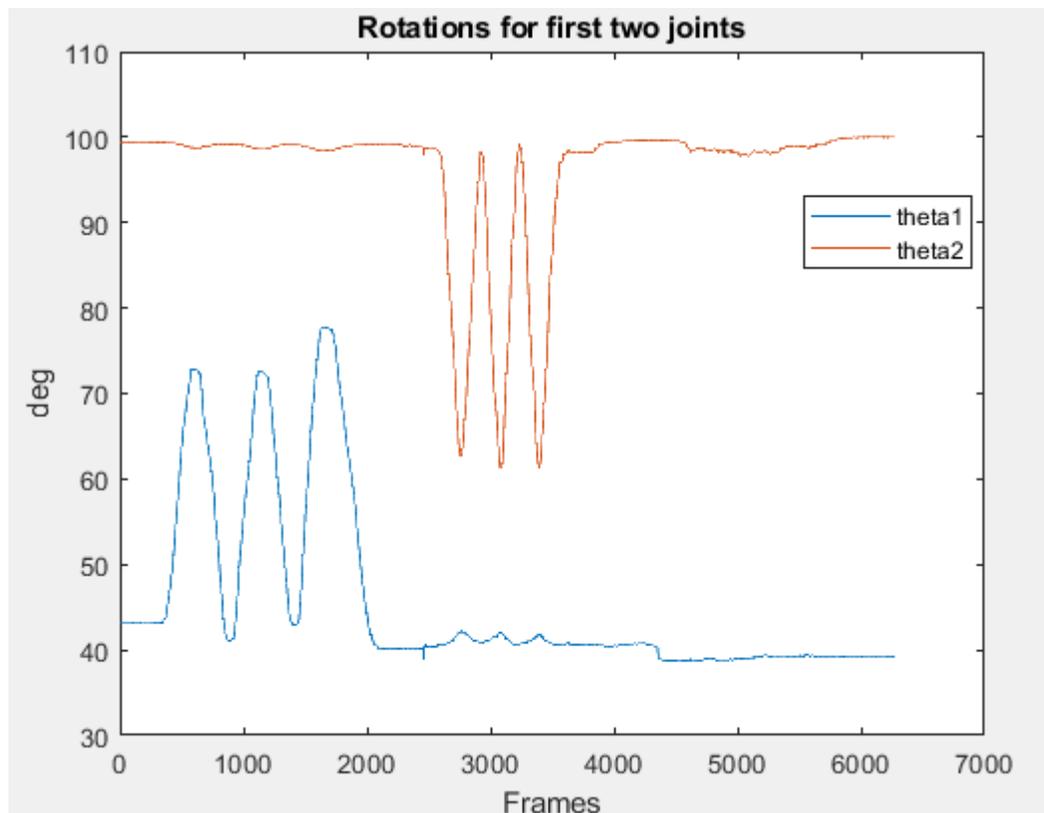


Figure 49: θ_1 and θ_2 for phantom arm (RRP case)

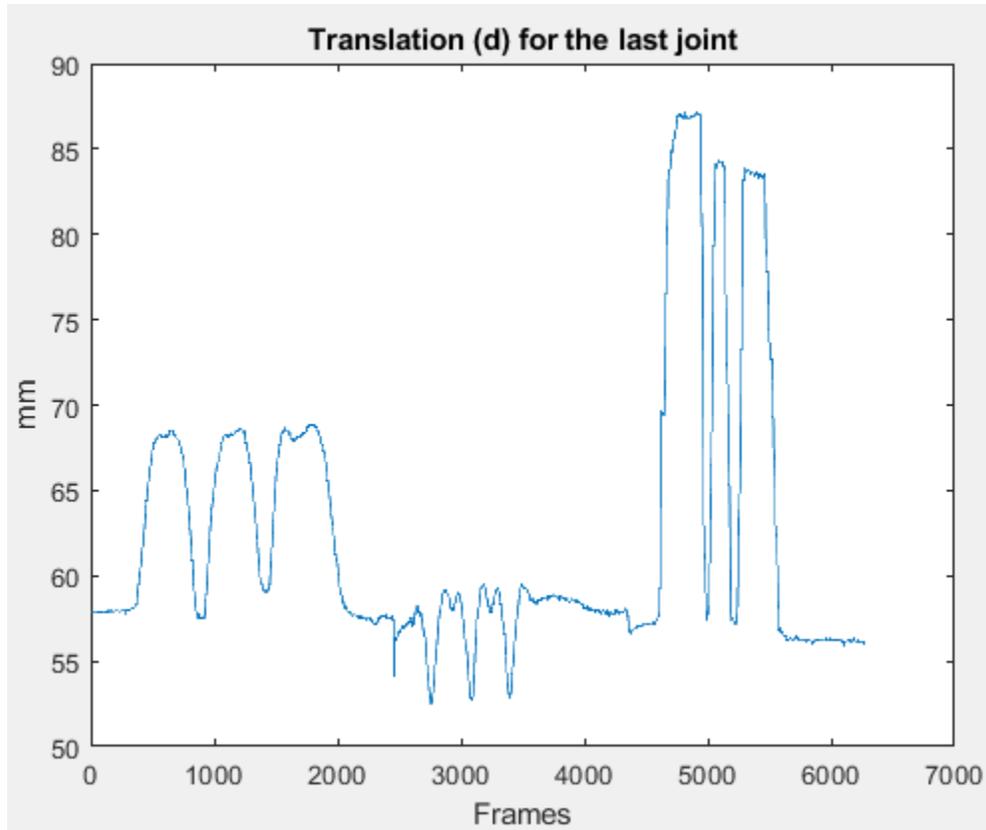


Figure 50: Translation 'd' for phantom arm (RRP case)

3.3.2 Biomechanics inverse kinematics approach

The initial thought was to simplify the kinematics of walking by resolving it to a simple 3 DOF, i.e. 3R planar manipulator configuration where the motion is along a plane perpendicular to the axes of rotation which is equivalent to the sagittal plane analysis. Three configuration space variables θ_1 , θ_2 and θ_3 corresponding to hip, knee and ankle angles would be calculated then for the model, assuming the joints to be revolute. It was decided to extend this approach to a higher degree of freedom model subsequently. The human leg in fact can be modeled as a 7-DOF manipulator since the hip and ankle joints are essentially constrained spherical wrists which have 3-DOF each, and the knee joint has 1-DOF. However, for modeling human motions, appropriate insights were gained leading to conclusion that a new approach would

have to be used for capturing biomechanics and the mentioned simplification would lead to inaccurate results. This has been further discussed in Chapter 4.

3.4 Custom Tracker Development

An application specific tracker was developed as a part of this work which would aid biomechanics motion capture.

3.4.1 Case and Cover Design

For the first prototype, the CAD model geometry was inspired by the commercial tracker and a cuff-based design was incorporated such that the developed tracker can be strapped on to an individual. Three hardware development kit components namely: EVM application board, the watchman core module, and the sensor breakout board were used for driving the case design. The EVM application board carries the GPIO ports, debugging setup and acts as a bridge to transfer data from the photodiodes to the PC. The watchman core module carries the high precision 48 MHz oscillators for recording the timestamps using the counters. It also carries the IMU. The sensor breakout board is where the photodiodes are connected to, using FFCs (flat flex connectors). The breakout board also includes 32 channel numbers which indicate which sensor is getting hit. The boards are powered using a 3.7 V, 900 mAh battery which also had to be accommodated inside the custom tracker. The design consisted of two parts: the case for accommodating the boards, the battery and the cover for placing the sensors. Figure 51 shows the design of the custom tracker:

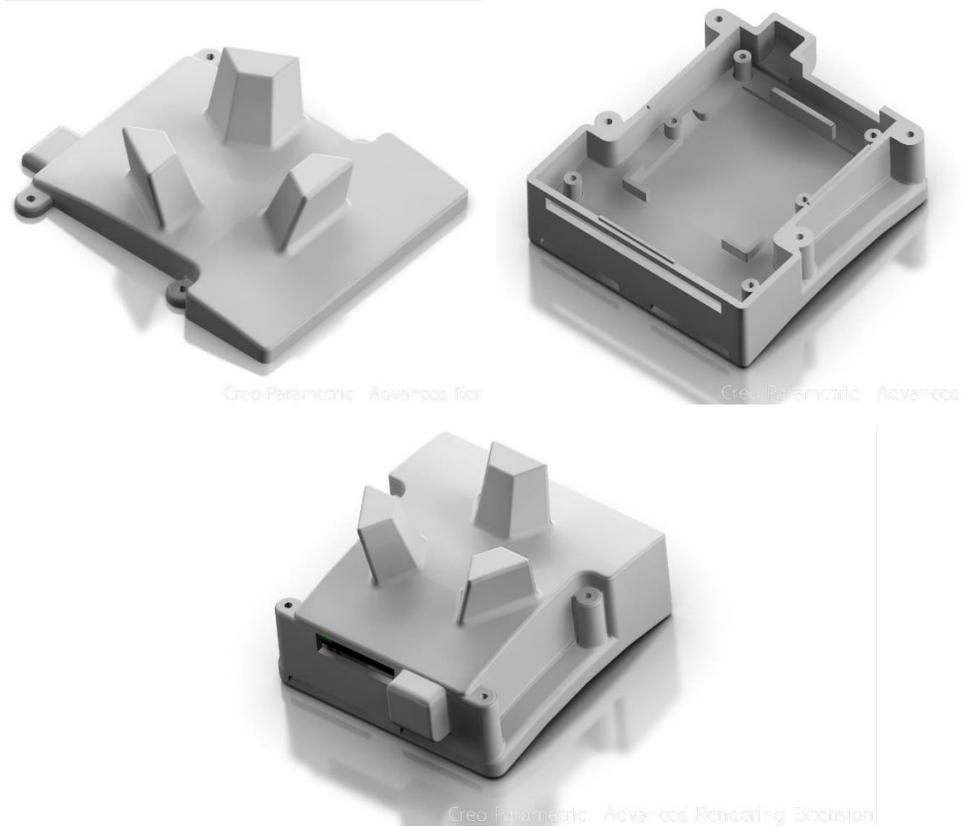
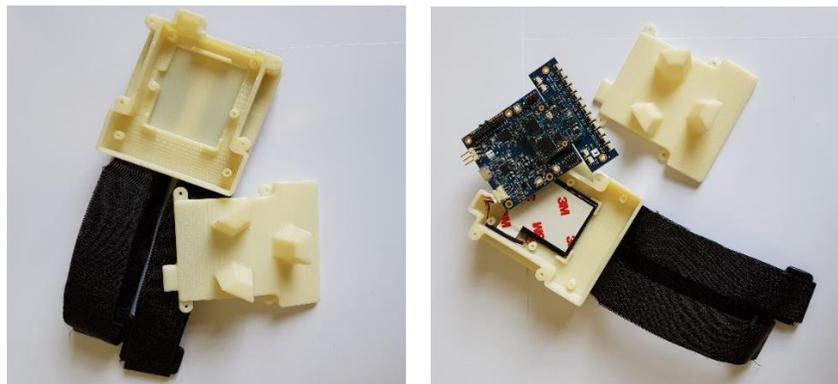


Figure 51: Render of CAD model for the case, cover, and assembly

The design was printed by using a 3D printer at UMass Amherst. The following figure shows the printed and assembled parts.



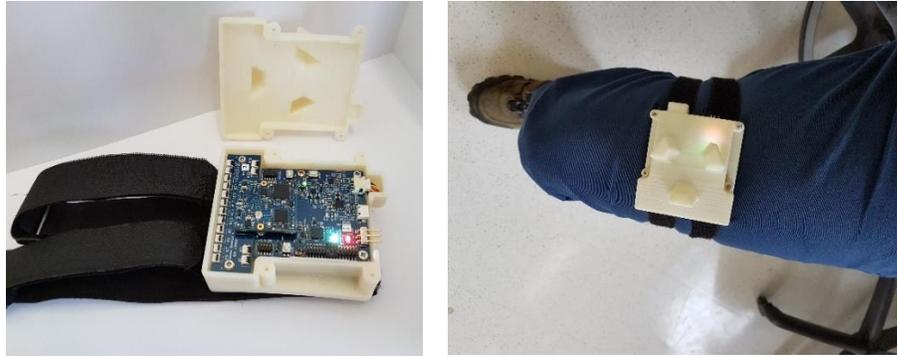


Figure 52: 3D Printed model and component assembly

3.4.2 IMU calibration

The IMU needed to be calibrated after mounting on the board inside the tracker. This can be done using a calibration jig or by placing the object in rectilinear positions on a flat surface and by taking six readings in +X, -X, +Y, -Y, +Z, -Z directions respectively. These readings output a calibration sphere for the IMU and generate offset values to reduce errors in calibration. The calibration routine of the IMU consists of taking readings for the accelerometer and gyroscope in the stated 6 directions and determining the variance and the mean for the samples. The calibration tool then outputs a relative fit error and generates biases or offset values to reduce those errors. These bias or offset values had to be included in the JSON file and uploaded to the object for each calibrated configuration.

The IMU calibration procedure was carried out for the designed custom tracker by placing the tracker inside a fairly rectilinear cubic box and holding the tracker in place using an adhesive. The hardware development IMU calibration tool was used to sample in 6 orthogonal directions. As suggested in the calibration routine; the combined fit error was less than the acceptable threshold of 0.1. The generated values were uploaded to the tracker in the JSON.

3.4.3 Sensor placement

The position is obtained using triangulation; an important parameter of which is the location of the sensors on the trackable object. The sensor locations can either be obtained using an optimal placement algorithm by using a developer tool called SteamVR Tracking HDK or manually based on personal preferences which can be evaluated as well using the same tool. The tool analyzes the geometry of the tracker and based on the number of required sensors, it runs simulations using up to 12 permutations that place the sensors at optimal locations to have an effective tracking performance and to reduce overall translational error and rotational error. A key parameter that determines the performance of the system is called the 'pose' which is a geometry that SteamVR constructs in real time based on which sensors are getting hit with the laser sweeps. SteamVR requires seeing a minimum of 5 sensors to snap on to a geometry comprised of the sensors to have an estimation of how the object is oriented or moving in space. The simulations require stereolithography (STL) file of the trackable object. Post-simulation, the tool outputs a number called the quality number which is a score out of 1000 with the lower number being an indication of more effective tracking and hence sensor placement. The simulation also outputs 4 plots in 2D as well as 3D which are a derivative of the tracking performance with respect to the base station. The plots show 4 evaluations based on the number of visible sensors, successful pose formation for boot up, translational errors and rotational errors. Any improvement in each individual evaluation is a function of overall tracking effectiveness. The procedure was carried out for 19 sensors and a quality factor of 317 was obtained.

The second way for placing the sensors is using the developer's preferred locations. This can be achieved by passing 3 vectors to the SteamVR software in a javascript object notation file (JSON). The first vector carries the sensor channel numbers which identifies the number associated with every sensor that gets hit. It mainly addresses the appropriate port number

on the FPGA. The second vector is the model points vector that carries the 3-dimensional coordinate information in (x, y, z) about the surface locations of the sensors in the designed CAD global frame. The final vector carries information about the model normals which indicate the direction in which the photodiode is facing. The model normal calculation requires two variables which are the model points and the distal points. Model points are the locations of the sensors on the tracker surface as mentioned previously. A corresponding distal point is a point in space perpendicular to the surface on which the model point is placed. Using the following relations, a model normal can be obtained for the respective model point:

$$x_{normal} = \frac{x_{distal} - x_{surf}}{\sqrt{(x_{distal} - x_{surf})^2 + (y_{distal} - y_{surf})^2 + (z_{distal} - z_{surf})^2}}$$

$$y_{normal} = \frac{y_{distal} - y_{surf}}{\sqrt{(x_{distal} - x_{surf})^2 + (y_{distal} - y_{surf})^2 + (z_{distal} - z_{surf})^2}}$$

$$z_{normal} = \frac{z_{distal} - z_{surf}}{\sqrt{(x_{distal} - x_{surf})^2 + (y_{distal} - y_{surf})^2 + (z_{distal} - z_{surf})^2}}$$

Another important parameter that needs to be included in the JSON file irrespective of the sensor placement technique is the location of the IMU in the object's coordinate frame which aligns the IMU's coordinate system with the objects coordinate system so that the visualized sensors match with the locations of the actual sensors. The procedure was followed for the designed tracker by choosing appropriate sensor locations based on the model geometry. The number of photodiodes was narrowed down from 26 to 19 eventually which is 4 less photodiodes than those on the commercial tracker.

Figure 53 shows the surface and distal points as well as the model normals placed for desired locations of the sensors.

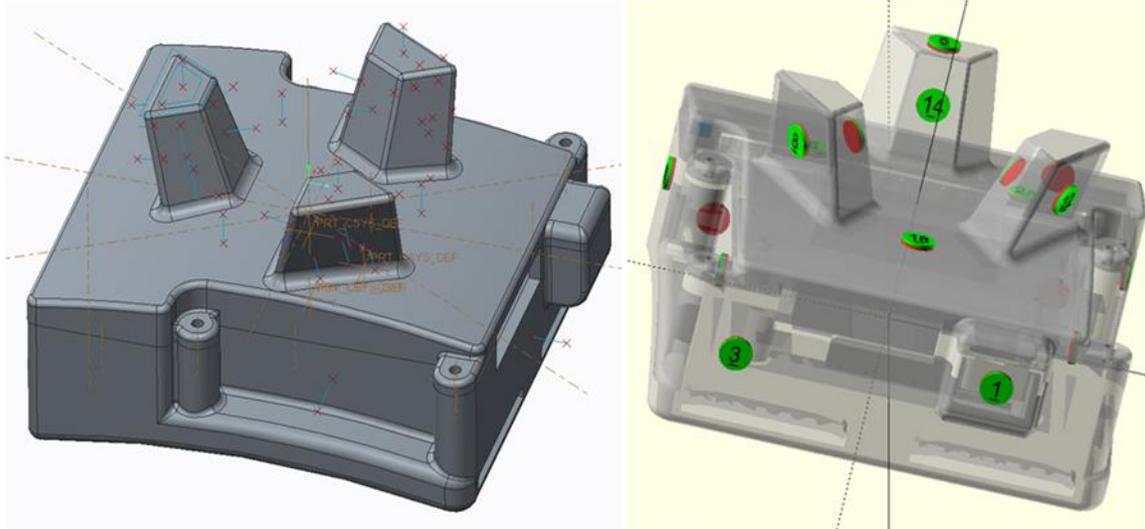


Figure 53: Manual Sensor Placement and sensor visualization tool

The photodiodes were placed on the desired locations using the manual placement technique for the first prototype; since some of the recommended locations by the optimal placement tool were impractical or could not be realized for the designed model. The sensors were mounted using heavy duty industrial double tape for the first prototype which would be replaced by adhesives in the future designs.

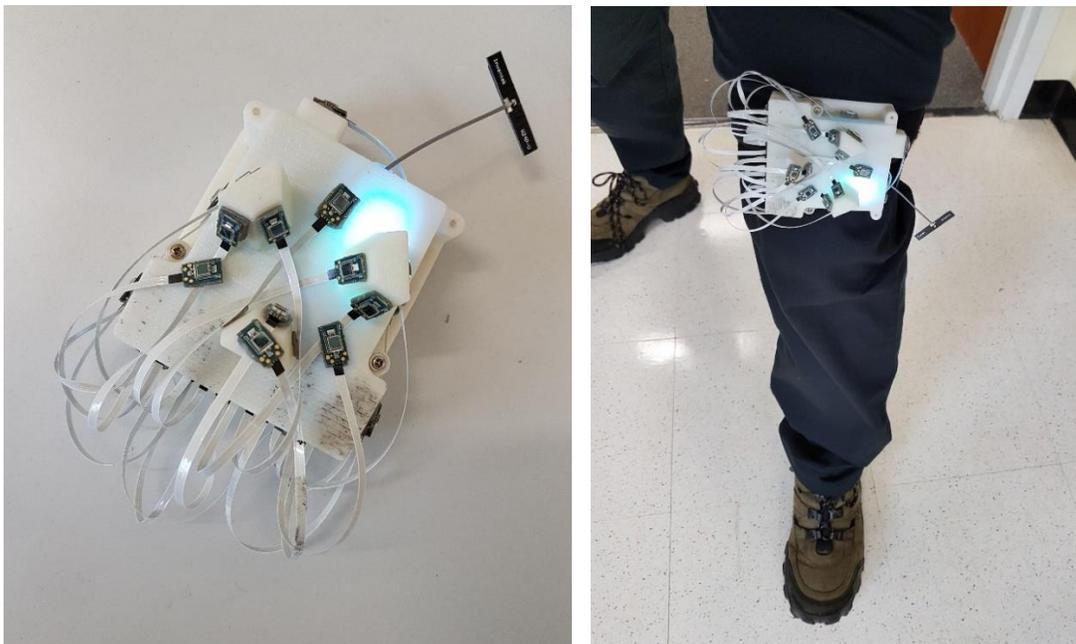


Figure 54: Custom tracker post optical sensor integration

The stated simulation procedure was carried out for the designed tracker. As seen in Figure 55, the simulation produces 4 distinct plots corresponding to 4 evaluation factors: number of visible sensors, initial pose, pose rotation and pose translation errors. The number of visible sensors is an indication of how many sensors a base station sees. Initial pose usually translates to the pose formation or object boot up. Translation errors result due to lack of planar baselines during pose formation which can be improved by increasing the width between the sensors. This increases the object's viewing distance from the base station. Similarly, pose rotation error is due to lack of depth baseline in pose between the resulting 3-sensor plane and the non-planar sensor. It should be noted that the following plots are a 2D representation of 3D spherical plots of the trackable object and fail to capture the entirety of solution. The red zones in the bottom region account for poor tracking or no tracking. But this is obvious since the object was mounted on a thigh and no optical data would be received on the underside.

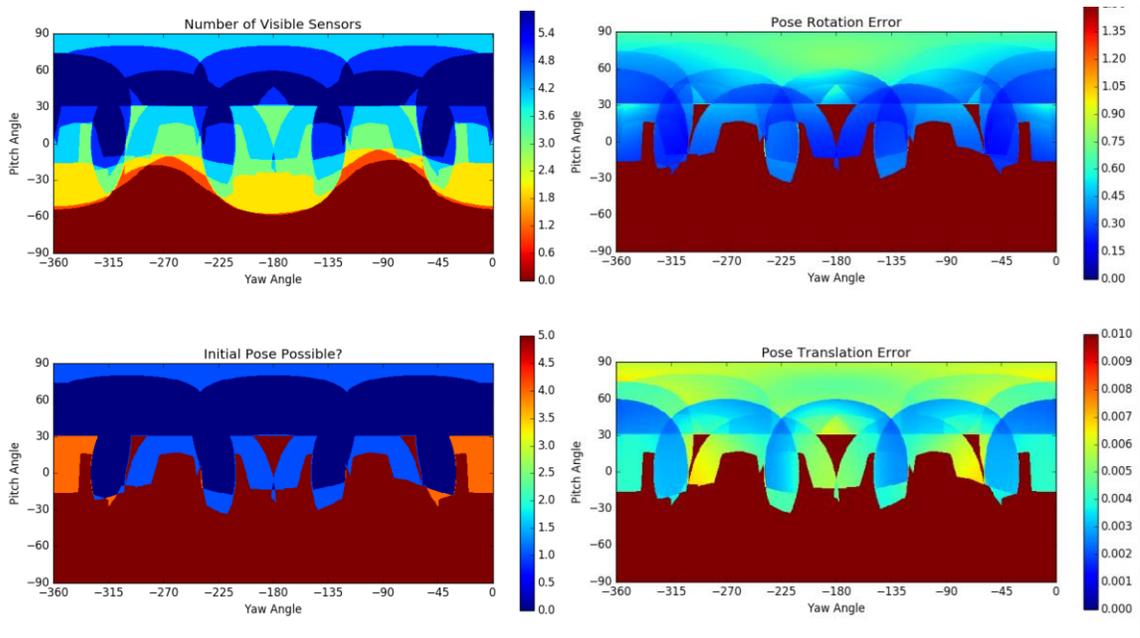


Figure 55: Simulation results using SteamVR Tracking HDK software for determining tracking performance using 19 sensors

3.4.4 Raw data acquisition

The IMU and optical sensor data were successfully obtained using the developed custom tracker.

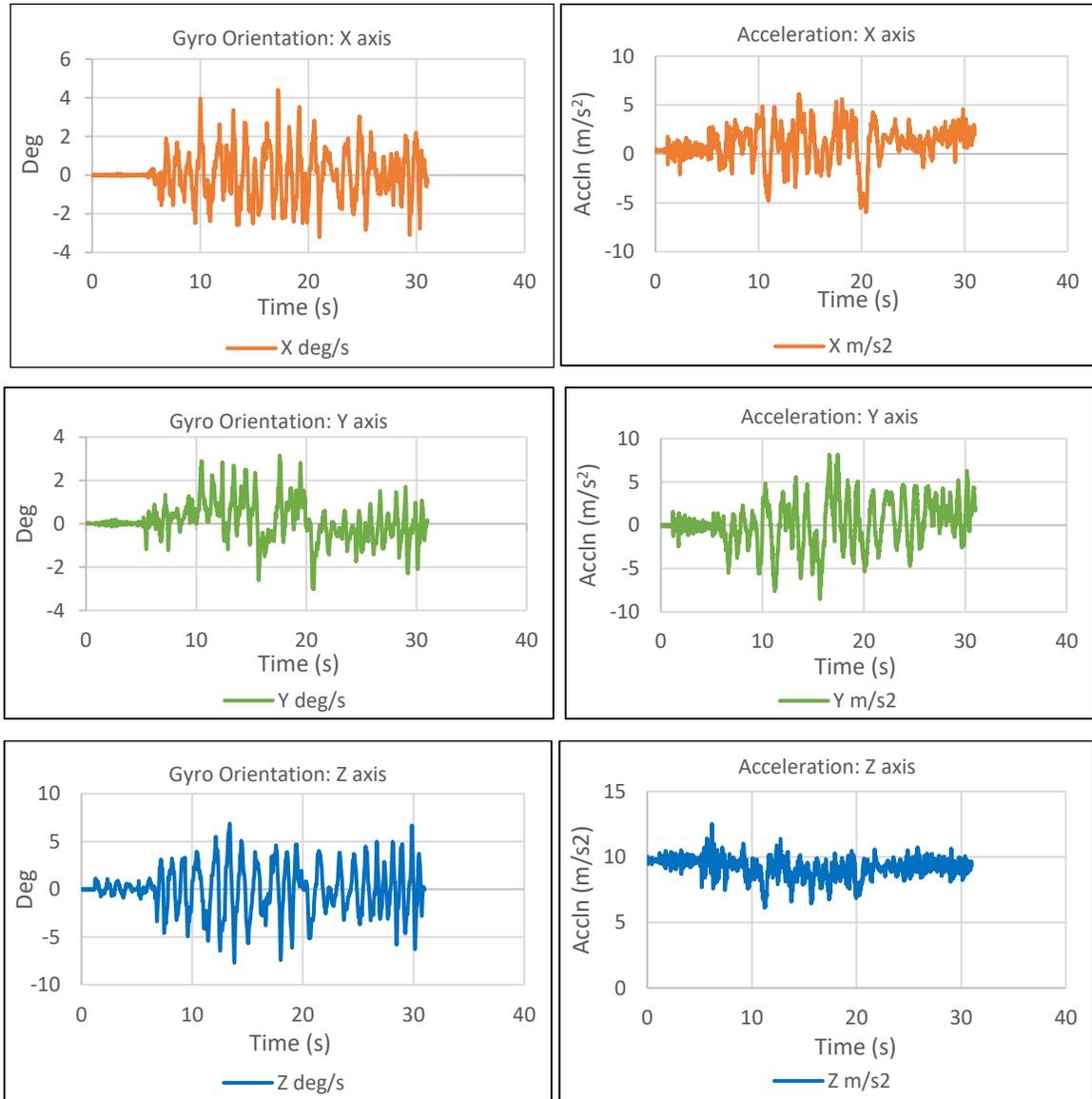


Figure 56: Gyroscope and Accelerometer raw data for X, Y and Z axes

The raw optical data as mentioned previously outputs the laser sweeps and the reference clock ticks and the sensor channel numbers to indicate which sensor is getting hit which is used for calculating the angle w.r.t the base station. It is also important to note that the angles

with respect to the base station obtained by the procedure stated in the working section in (Chapter 2) Background would be very close to each other. Hence, we require high resolution and high precision counters for the timestamps which can record individual events for different sensors for even relatively small angles. This speculation was observed in the following sampled data.

Figure 57 shows the summary of the raw optical data. The plot portrays which sensor channel is getting hit at a time instant for an overall sampling period of about 35s. Since we are sampling at 60 Hz the continuous lines on the plot are actually independent sensor hits but plotting for a lower resolution we obtain continuous lines. 19 sensors were used for the developed tracker connected at the following channel numbers: (0, 1, 3, 5, 7, 8, 10, 12, 14, 15, 16, 18, 20, 21, 22, 23, 24, 26, 27)

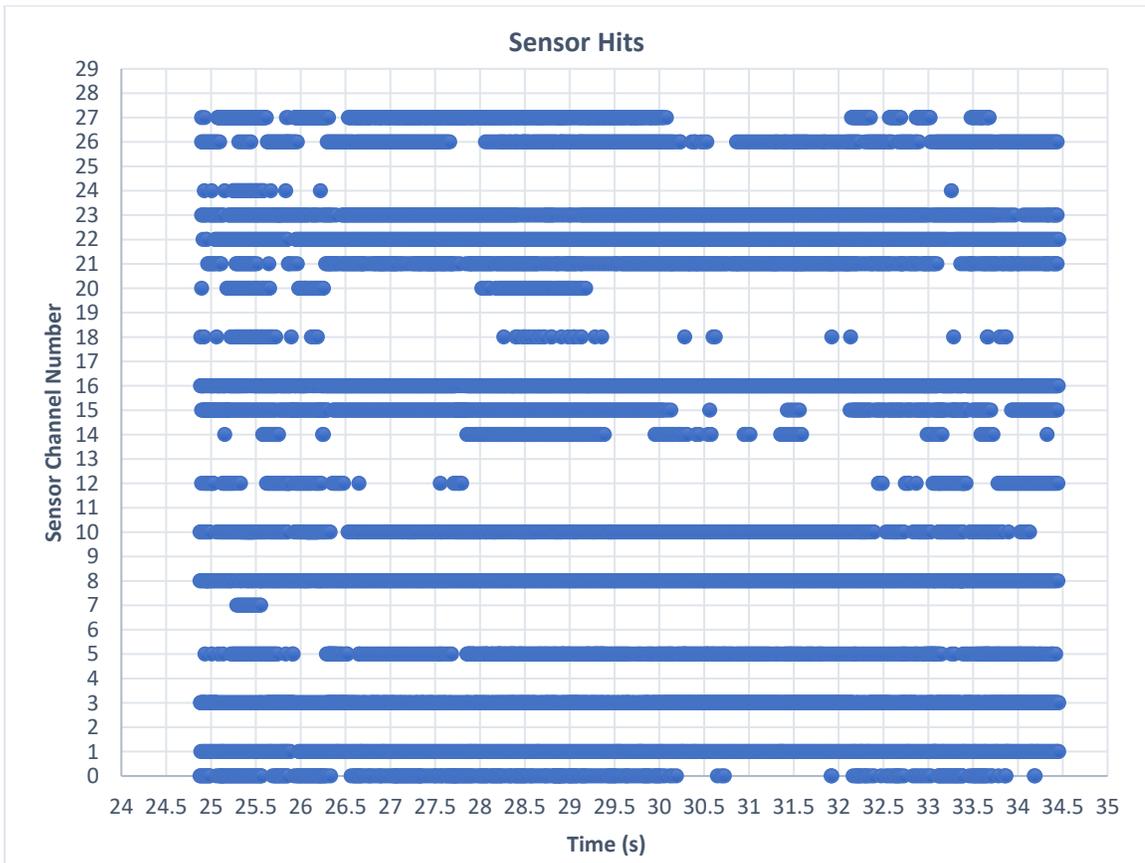


Figure 57: Optical sensor raw data for designed tracker

3.4.5 Optical sensor recalibration

Optical sensor recalibration is a procedure of reducing the sensor placement errors using the VR Tracking Calibration tool. It primarily determines how far-off are the photodiodes or optical sensors in terms of their actual locations as opposed to the theoretical locations in the JSON file. The recalibration routine has to be carried out after initial sampling to improve tracking performance. The output of the calibration routine are the updated values for the model normals and the model points which represent the actual locations of the sensors accurately. The recalibration routine was followed for the designed tracker to obtain the updated model normals and model points which were uploaded to the tracker through the JSON file. 200 hits were taken per sensor by moving the designed tracker in the trackable space and rotating and translating it throughout the routine to get sufficient hits. The procedure was repeated three times with each iteration reducing the net error thus improving overall tracking performance.

CHAPTER 4

DISCUSSION AND FUTURE WORK

4.1 Custom Tracker Optimization

The tracking performance for the developed custom tracker was evaluated based on the simulations regarding sensor placement and by using the visualization interface developed by FTL. Decent tracking was observed using the developed GUI with the custom tracker.

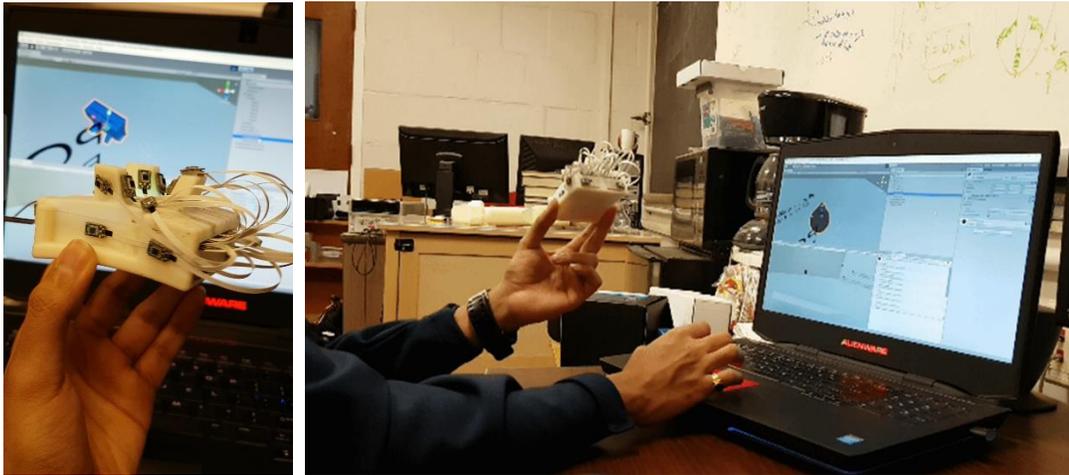


Figure 58: Real-time tracking using the custom tracker

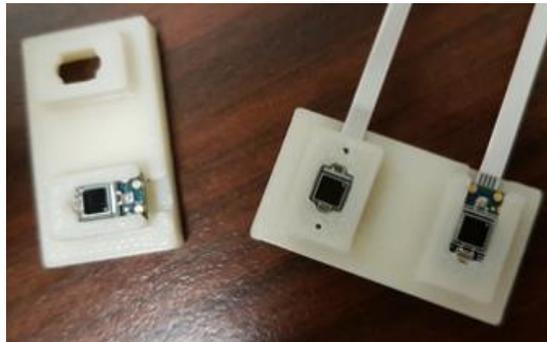


Figure 59: Pocket designs to hold photodiodes

It is important to note that since the sensors were double-taped to their locations, the theoretical locations were not precisely satisfied. Hence, there was a need of repeating the recalibration procedure thrice for decent tracking performance. An ideal design would be to mount the sensors on their exact location. It is necessary to have the centroid of each

photodiode which is about 2.65 X 2.65 mm area aligned exactly with the obtained model normal in the JSON file. This can be ensured using pockets integrated within the tracker geometry that ensure accurate placement. A few pocket designs were experimented as shown in Figure 59 which can be implemented for subsequent iterations.

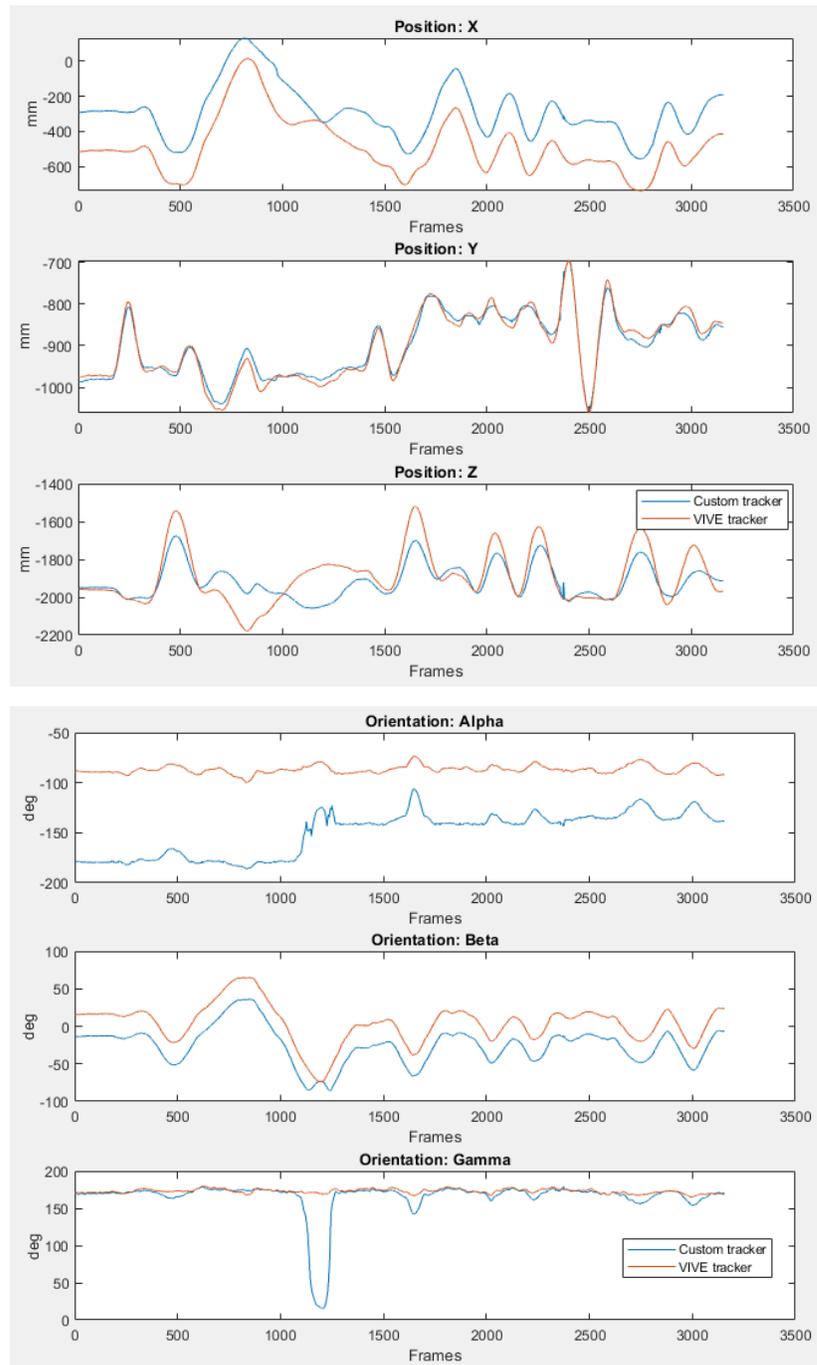


Figure 60: Custom tracker position and orientation plots

Figure 60 shows the experimental data collected while tracking using the commercial VIVE tracker and the developed custom tracker. A reference tracker was not used for tracking this time. Both trackers were kept on a flat surface which was moved in space to record motion and capture poses with respect to the base station. The idea here was not to quantify the accuracy of tracking; but rather compare the overall tracking performance. As seen in the plots, there wasn't any significant tracking loss and the developed custom tracker tracked relatively well with 4 less photodiodes. Further studies will also involve comparing the accuracy of the developed prototypes with the VIVE trackers.

In terms of circuit design, it is possible to strip down the hardware development kit of its expansion ports and additional pins. This will reduce the internal footprint thus reducing the case size and mass. The current custom tracker prototype weighs 106 g which is 16 g heavier than the VIVE tracker. As mentioned previously a smaller footprint would be desirable for the tracker for body or gait tracking since that would reduce the tissue artifacts and the inertia mass effects leading to better results for kinematic studies or analysis. Skin tissue artifacts are common types of issues associated with motion capture. These artifacts can greatly influence the results leading to inaccuracies that might range upto 40 mm for position and about 20 deg for orientation [40][41]. One can also reduce the number of sensors by using multiple base stations perhaps more than 4 such that the minimum number of photodiodes are always visible and any possibility of an occlusion from one base station would be handled by the other. Moreover, other pose estimation techniques or algorithms could be implemented to reduce the sensor numbers [42].

4.2 Cross-validation against camera-based motion capture

Cross-validation can also be carried out based on the available Qualisys motion capture system at UMass Amherst and the lighthouse technology sampling at the same time. Figure

61 shows the setup for the cross-validation. This was tried out; however, since both the technologies use infrared light sources, a considerable amount of crosstalk was observed when the systems were sampling at the same time. It is possible to use non-infrared lit markers or active marker systems for motion capture which produce their own luminescence to avoid the cross-talk issue. Another way is to adjust the systems such that they sample at a phase offset using discrete flashes of light instead of continuously flooding the room with IR-light. This will have to be explored for any further cross-validation, especially when the motion cannot be replicated for instance the human motion.

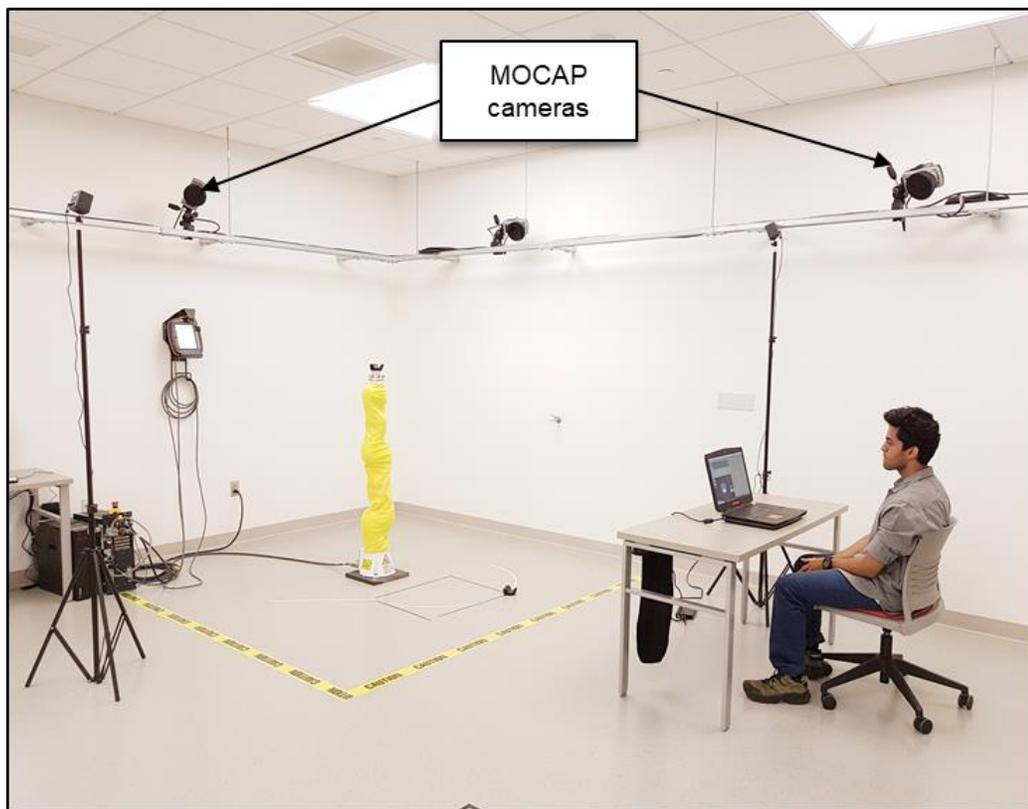


Figure 61: Setup for cross-validation against camera-based motion capture

4.3 Biomechanics motion capture and gait analysis

The approach described in the section 3.3.1 is valid for rigid bodies and for purely revolute joints. As for biomechanics motion capture, inverse kinematics needs to be performed based

on numerical methods employing optimization for the best model fit on experimental data. In case of human motion analysis, the body segments are a combination of rigid and elastic elements. The human joints as well, are a combination of rotational and translational elements or roller joints, hence assuming a model with perfectly revolute joints would lead to inaccurate results. Similarly, for lower limb prosthesis due to the elastic foot and the nature of socket and residual limb interface, rigid body assumptions won't be valid. In case of an amputee; various parametric relations in terms of translations and rotations between the prosthetic components would provide useful insights to the prosthetist on prosthetic alignment and can further influence redefining a currently adopted technique. These joint angle relations can be determined using inverse kinematics. This inverse kinematics approach is based on minimizing the error between the model and experimental data.

For capturing human motions, trackers would be mounted on the pelvis, thigh, shank and shoe as shown in Figure 62. Additional trackers might be used for localizing the model and for ground reference tracking. Using the lighthouse technology, homogeneous transformations will be obtained for each of the trackers as well as the model frame with respect to the global frame. Further transformations will be calculated from the model frame to each tracker frame.

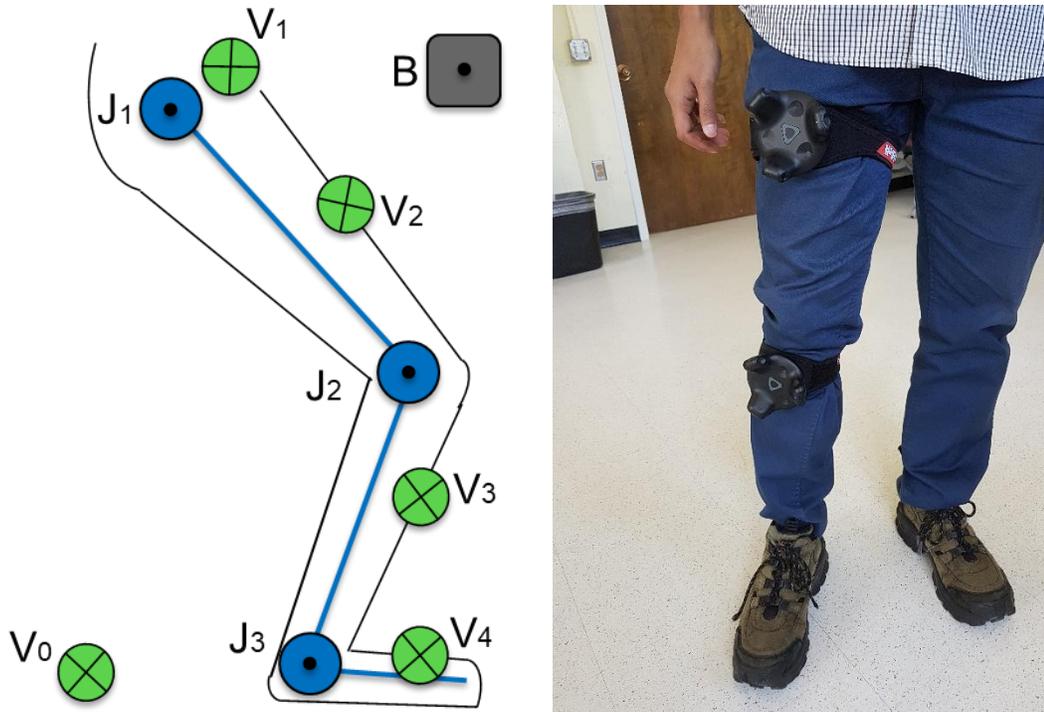


Figure 62: Setup for biomechanics inverse kinematics

It is important to create a decent model for tracking the human limb. For doing this, static trials were performed using the marker-based motion capture system at Umass Amherst where retroreflective markers were placed on the bony prominences of one of the limbs to localize ankle, knee and pelvis. Markers were also placed on the thigh. It would also be important to know where the trackers would be placed when the lighthouse technology would be used for testing. This was done by placing a spare marker in the model and to keep a reference for mounting the trackers. Further work will involve scaling this model using OpenSim which is a musculoskeletal modeling software. The model will help determine the joint axes and the joint centers as well as the offsets induced by the surface trackers.

The desired position at each frame will be obtained from the experimental data of the trackers in model frame:

$$P_d(i) = [X_d(i) \quad Y_d(i) \quad Z_d(i)]^T$$

The current position: $P_c = f(q_i)$ which would be determined from the forward kinematics relationship would be generated as a function of joint angles. For each frame, we can choose an initial guess for each angle depending on the start position and based on joint constraints which would be the joint limits for depending on the nature of the joint. These also help to avoid kinematics redundancies [43].

$$q_i < [Lower\ limit(i),\ Upper\ limit(i)]$$

Starting with an initial guess for q_i and substituting in the model we have at each frame:

$$P_c(i) = [X_c(i)\ Y_c(i)\ Z_c(i)]^T$$

An objective function would be formulated based on the root mean squared error minimization between the model and experimental pose data at each frame and the joint angles will be calculated. The optimization routine will be based on gradient descent algorithm or similar approaches.

Minimize the summed RMS error:

$$F = \min. ||P_d(i) - P_c(i)||^2$$

$$F = \sqrt{[X_d(i) - X_c(i)]^2 + [Y_d(i) - Y_c(i)]^2 + [Z_d(i) - Z_c(i)]^2}$$

OpenSim also has a dedicated inverse kinematics tool which performs the above procedure when the model and experimental data are provided as inputs. It uses relationships or functions for joints rather than assuming them to be revolute which would lead to accurate results. However, currently it is best suited for optical tracking using the marker-based motion capture where it uses the localization marker clusters around a joint centre to perform the above procedure. Using the trackers however, we have one pose per tracker per joint or segment and the localization would have to be explored. One possibility is creating virtual markers using the tracker data and weighing them accordingly. FTL is currently

working on a plugin for OpenSim that would work with the tracker data. Hence, human motion quantification would be carried out in the future.

Further studies also involve pairing up the system with a pressure pad mat or force plate to obtain the ground reaction forces. This data would be critical for calculating joint moments and forces using inverse dynamics. Joint velocities can be also calculated by computing Jacobians from the joint angle data obtained using inverse kinematics.

CHAPTER 5

CONCLUSION

From the obtained results, it can be established that the system can be applied to scientific studies where precision motion capture is desired. The technology is not limited to the recreational domain and other motion tracking applications can be realized using this system. The studies carried out show that data acquisition is possible for research work without the obligation of any virtual reality elements. Issues as in section 3.2.1 and mentioned in other studies [31][32] were encountered, but it was possible to resolve them with the resources in hand without any major system changes.

The system is precise upto millimeter level for tracking positions while sub-degree level precision was obtained for orientations based on the repeatability studies carried out. Provided that good tracking is ensured, then repeatability values obtained for position are on a sub-millimeter level as well. As for accuracy, by cross-validating the system against that of an industrial robotic arm, it was shown that millimeter accuracy is possible for recording positions and sub-degree level accuracy can be obtained for tracking orientations. It was further possible to quantify angles based on tests carried on the phantom limb. Implications were also drawn for expansion of the system for gait and biomechanics capture for applications in the medical field. Hence, this technology can serve as a cost-effective alternative to current commercially available solutions.

REFERENCES

- [1] Ziegler-Graham K, MacKenzie EJ, Ephraim PL, Travison TG, Brookmeyer R. Estimating the Prevalence of Limb Loss in the United States: 2005 to 2050. *Archives of Physical Medicine and Rehabilitation* 2008;89(3):422-9.
- [2] Owings M, Kozak LJ, National Center for Health S. *Ambulatory and Inpatient Procedures in the United States, 1996*. Hyattsville, Md.: U.S. Dept. of Health and Human Services, Centers for Disease Control and Prevention, National Center for Health Statistics; 1998.
- [3] *Amputee Statistics you ought to know*, Advanced Amputee Solutions, LLC., 2012, <<https://www.advancedamputees.com/amputee-statistics-you-ought-know>>
- [4] *Common Prosthetic Issues*, Medical Center Orthotics and Prosthetics, 2017
- [5] C. H. Lloyd, S. J. Stanhope, I. S. Davis, and T. D. Royer, "Strength asymmetry and osteoarthritis risk factors in unilateral trans-tibial, amputee gait," *Gait Posture*, vol. 32, no. 3, pp. 296–300, 2010. <<http://dx.doi.org/10.1016/j.gaitpost.2010.05.003>>
- [6] A. K. Silverman, N. P. Fey, A. Portillo, J. G. Walden, G. Bosker, and R. R. Neptune, "Compensatory mechanisms in below-knee amputee gait in response to increasing steady-state walking speeds," *Gait Posture*, vol. 28, no. 4, pp. 602–609, 2008.
- [7] Kobayashi, T., Arabian, A. K., Orendurff, M. S., Rosenbaum-Chou, T. G., & Boone, D. A. (2014). Effect of alignment changes on socket reaction moments while walking in transtibial prostheses with energy storage and return feet. *Clinical Biomechanics*, 29(1), 47–56. <<https://doi.org/10.1016/j.clinbiomech.2013.11.005>>
- [8] Boone, D. A., Kobayashi, T., Chou, T. G., Arabian, A. K., Coleman, K. L., Orendurff, M. S., & Zhang, M. (2013). Influence of malalignment on socket reaction moments during gait in amputees with transtibial prostheses. *Gait and Posture*, 37(4), 620–626. <<https://doi.org/10.1016/j.gaitpost.2012.10.002>>
- [9] Kobayashi, T., Orendurff, M. S., Zhang, M., & Boone, D. A. (2015). Socket reaction moments in transtibial prostheses during walking at clinically perceived optimal alignment. <<https://doi.org/10.1177/0309364615588345>>
- [10] C. R. E. Duhaylungsod, C. E. B. Magbitang, J. F. I. R. Mercado, G. E. D. Osido, S. A. C. Pecho, and A. R. dela Cruz, "Detection of gait abnormality through leg symmetry and temporal parameters," 2017 IEEE 9th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag., pp. 1–4, 2017. <<http://ieeexplore.ieee.org/document/8269519/>>
- [11] R. Mooney et al., "Radiographic parameters improve lower extremity prosthetic alignment," *J. Child. Orthop.*, vol. 7, no. 6, pp. 543–550, 2013.
- [12] *Sensorized Prosthetic Alignment ReadOut*, FTL Labs Corp., 2017
- [13] A. K. T. Ng, L. K. Y. Chan, and H. Y. K. Lau, "A low-cost lighthouse-based virtual reality head tracking system," 2017.
- [14] Lin MC, Wu Y, Edwards M. Vertical alignment axis for transtibial prostheses: A simplified alignment method. *J. Formos Med Assoc* 2000;99(1):39-44

- [15] Zahedi MS, Spence WD, Solomonidis SE, Paul JP. Alignment of lower-limb prostheses. *J Rehabil Res Dev* 1986;23(2):2-19.
- [16] S. Sin, "A new alignment jig for quantification and prescription of three-dimensional alignment for the patellar-tendon-bearing trans-tibial prosthesis," *Prosthetics Orthot.*, vol. 23, no. 3, pp. 225–230, 1999.
- [17] L.A.S.A.R. Posture, Product Details, Ottobock, 2016 <<https://professionals.ottobockus.com/Prosthetics/Materials/Equipment/Alignment-and-Measuring/Laser-Technology/L-A-S-A-R-Posture/p/743L100~5110>>
- [18] A. J. Ikeda, K. D. Reisinger, M. Malkush, Y. Wu, M. L. Edwards, and R. S. Kistenberg, "Á priori alignment of transtibial prostheses: A comparison and evaluation of three methods," *Disabil. Rehabil. Assist. Technol.*, vol. 7, no. 5, pp. 381–388, 2012.
- [19] A. Cappozzo, U. Della Croce, A. Leardini, and L. Chiari, "Human movement analysis using stereophotogrammetry. Part 1: Theoretical background," *Gait Posture*, vol. 21, no. 2, pp. 186–196, 2005.
- [20] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate $O(n)$ solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, 2009.
- [21] A. Leardini, C. Belvedere, F. Nardini, N. Sancisi, M. Conconi, and V. Parenti-Castelli, "Kinematic models of lower limb joints for musculo-skeletal modelling and optimization in gait analysis," *J. Biomech.*, vol. 62, pp. 77–86, 2017.
- [22] M. Donati, V. Camomilla, G. Vannozzi, and A. Cappozzo, "Anatomical frame identification and reconstruction for repeatable lower limb joint kinematics estimates," *J. Biomech.*, vol. 41, no. 10, pp. 2219–2226, 2008.
- [23] E. L. Erickson, C. Leydon, and S. L. Thibeault, "Accuracy and Precision of a Custom Camera-Based System for 2-D and 3-D Motion Tracking During Speech and Nonspeech Motor Tasks," *J. Speech, Lang. Hear. Res.*, vol. 57, no. June, pp. 1679–1691, 2014.
- [24] Lang, Ben (2017-04-13). "Latest HTC Vives Are Shipping with Tweaked Base Stations, Redesigned Packaging – Road to VR". *Road to VR*. Retrieved, Nov. 2017
- [25] SteamVR. SteamVR Tracking HDK Documents. Valve Corporation, 2017. <[https:// partner.steamgames.com](https://partner.steamgames.com)>
- [26] Allyn Malventano, SteamVR HTC Vive In-depth - Lighthouse Tracking System Dissected and Explored, General Tech, 2016.
- [27] S. Briot and W. Khalil, "Dynamics of Parallel Robots," vol. 35, pp. 19–33, 2015.
- [28] Uicker JJ, Jr., Denavit JJ, Hartenberg RS. An Iterative Method for the Displacement Analysis of Spatial Mechanisms. *ASME. J. Appl. Mech.* 1964;31(2):309-314. doi:10.1115/1.3629602.
- [29] M. W. Spong, S. Hutchinson, and M. Vidyasagar, "FORWARD KINEMATICS: THE DENAVIT-HARTENBERG CONVENTION BT - Robot Dynamics and Control," *Robot Dyn. Control*, pp. 1–32, 2003.
- [30] R. Balasubramanian, "The Denavit Hartenberg Convention," pp. 1–6, 2013.

- [31] D. C. Niehorster, L. Li, and M. Lappe, "The accuracy and precision of position and orientation tracking in the HTC vive virtual reality system for scientific research," *Iperception.*, vol. 8, no. 3, pp. 1–23, 2017.
- [32] 'Position Drift after tracking lost (Without HMD)', SteamVR Tracking HDK; General Discussions, 2017
<<http://steamcommunity.com/app/507090/discussions/0/1489992080523267989/>>
- [33] G. G. Slabaugh, "Computing Euler angles from a rotation matrix," denoted as TRTA Implement. from <http://www.starfireresearchcomservicesjava3dsamplecodeFlorinEulers.html>, vol. 6, no. 2000, pp. 1–6, 1999.
- [34] KUKA, "LBR iiwa," Pub Spez LBR iiwa en, pp. 1–69, 2016.
- [35] D. Busson, R. Bearee, and A. Olabi, "Task-oriented rigidity optimization for 7 DOF redundant manipulators," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14588–14593, 2017.
- [36] Y. R. S. Urz, L. M. Affolter, and R. S. Smith, "Parameter Identification of the KUKA LBR iiwa Robot Including Constraints on Physical Feasibility," *IFAC Pap.*, vol. 50, no. 1, pp. 6863–6868, 2017.
- [37] M. Granja, N. Chang, V. Granja, M. Duque, and F. Llulluna, "Comparison between Standard and Modified Denavit-Hartenberg Methods in Robotics Modelling," vol. 1, no. 1, pp. 1–10, 2016.
- [38] J. J. Craig, *Introduction to Robotics*. Addison Wesley, 1986
- [39] P. I. Corke, "A Simple and Systematic Approach to Assigning Denavit–Hartenberg Parameters," *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 590–594, 2007.
- [40] T. Ryu, "Application of Soft Tissue Artifact Compensation Using Displacement Dependency between Anatomical Landmarks and Skin Markers.," *Anat. Res. Int.*, vol. 2012, p. 123713, 2012.
- [41] U. Della Croce, "Soft tissue artifacts in human movement analysis," *Proc. 9th Symp. 3D Anal. Hum. Mov.*, no. May, 2006.
- [42] Y. Yang, D. Weng, D. Li, and H. Xun, "An improved method of pose estimation for lighthouse base station extension," *Sensors (Switzerland)*, vol. 17, no. 10, pp. 1–14, 2017.
- [43] S.-H. Cha, T. A. Lasky & S. A. Velinsky (2006) Kinematic Redundancy Resolution for Serial-Parallel Manipulators via Local Optimization Including Joint Constraints, *Mechanics Based Design of Structures and Machines*, 34:2, 213-239, doi: 10.1080/15397730600778527