

# An Inverse Kinematics Model for the lower body Motion: Implementation of the Fabrik-Algorithm as part of a Motion tracking Pipeline



Hochschule für Angewandte Wissenschaften  
Hamburg  
Fakultät Design, Medien und Information  
Department Medientechnik

Bachelor-Thesis

zur Erlangung des akademischen Grades B.Sc.

Wladimir Nabok



Erstprüfer: Prof. Dr.-Ing. Katrin Wolf

Zweitprüfer: Jens Reinhardt

Hamburg, 29.04.2019

## Table of Contents

|   |    |
|---|----|
| Abstract .....  | 2  |
| Introduction .....  | 2  |
| Related Works .....   | 3  |
| Theory .....  | 3  |
| Planes of Motion .....  | 3  |
| Kinematics .....  | 5  |
| Forwarded Kinematics .....  | 5  |
| Inverse Kinematics .....  | 5  |
| Leg Model .....   | 5  |
| Method .....  | 10 |
| Motion Tracking .....   | 10 |
| Process .....   | 10 |
| Design of the Motion Set .....  | 11 |
| Motion Tracking Skeleton Model .....                                    | 12 |
| Data Export .....   | 14 |
| Infrastructure and Solving Application .....                            | 15 |
| Mokka Motion Analyzer .....   | 17 |
| Problem and Solutions .....   | 19 |
| Result .....  | 21 |
| Visual Analysis .....   | 21 |
| Jogging Motion in Different Variants .....                              | 22 |
| Trajectory Error of Motions .....                                       | 24 |
| Case: Treadmill Data-set, Constrained, Knee Joint .....                 | 24 |
| Case: Treadmill Data-set, Constrained, Heel Joint .....                 | 26 |
| Case: Jumping Jacks Data-set, Constrained, Knee Joint .....             | 27 |
| Case: Jumping Jacks Data-set, Constrained, Heel Joint .....             | 29 |
| How is this Result to be Interpreted? .....                             | 30 |
| Discussion .....  | 32 |
| The Model .....   | 32 |
| The Algorithm .....   | 32 |
| Infrastructure .....  | 33 |
| Conclusion and Further work .....                                       | 34 |
| Appendix A: Parameters for the Inverse Kinematics Solving Program ..... | 35 |
| Appendix B: Rizzoli Marker Descriptions [15] .....                      | 36 |
| Appendix C: Measurement Tables .....                                    | 37 |
| Table of Source .....   | 46 |
| Table of Figures .....  | 48 |

## Abstract

In this thesis a custom-made benchmark environment for a humanoid lower-body model and inverse kinematics algorithm shall be implemented. It can then be used to compare metrics of the execution of the algorithms and the accuracy of the models.

Motion tracking technology is used to furthermore get position measurements of lower body markers and map them to the joints set of the researched model. Comparing the motion tracking measurements with the trajectories of joints calculated by the specified algorithm will be the core of this thesis.

## Introduction

Motion tracking is a technology, that all allows tracking capturing motion, digitalizing it and save it on a computer. There are different technologies on how to achieve that, the most widespread one being marker based. As this technology is used in different fields like medicine, entertainment and virtual reality, advancements in this technology, will also advance the fields, that use the technology.

Narrowing down the motions that can be tracked involving the humanoid lower body movement we will be closer to the topic of this thesis. Lower-body motion involves every movement and rotation of the hip, waist, thigh, foot, ankle, toe tip, legs, so a not small amount of body parts. Example for motions can be a dance step, a gait, a kick, a jog and many more.

Before one can capture these movements, one must go through one of the procedures of motion capture, that is the placement of trackers on a motion tracking suit, which is a suit optimized for the motion tracking technology. A lower-body marker-set can involve the placement of 20 markers on the specified places for a decent accuracy. This might be a process that has to be only once per tracking session, however if there would be a way to use the less markers but get the same result, I would argue people would use this solution to cut time cost of the process.

Also, there is consumer grade motion tracking involved in the currently growing virtual reality technology. There we have the case, where setup procedures want to be minimized for users to have a low entry barrier to put on the entertainment device. So, a different way of tracking movement must be modelled, in order to see human representations of their selves, avatars with actual limbs.

There has been research done to these problems and this thesis shall serve as a first benchmark, that implements variations of leg models and one inverse kinematics algorithm.

## Related Works

Animation of legged figures are not a new research topic, researchers have tried to model leg motion, gait cycles in particular. Interpolation via keyframes has been a widely used tool back then as well. Since this method misses on constraints, interpolated positions can occur in unwanted places, like the ground. To resolve these difficulties forward kinematics has been used to move a chain of joints in the wanted direction, with specified joint angles and link lengths forward. Its calculation has been based on Jacobian matrices. A successful model could be deduced. [1]

The trajectory of a leg model with the use of a Jacobean based inverse kinematics algorithm has been calculated, with the inclusion of angle constraints, that incorporate comfort zones of a humanoid body. A genetic algorithm has been used for the optimization problem. [2]

The requirements for a leg model can vary. While in computational science position and length of joints is enough, in order to create an even more realistic model of humanoid locomotion a musculotendon model can be incorporated. This way muscle states can be predicted, and energy input and output can be simulated. Such a model is targeting a biologically motivated parametrization. [3] [4]

As inverse kinematics algorithms lay the basis for this thesis, having an overview of existing algorithms is important. They can be categorized as analytical, numerical and data-driven.

A popular numerical method is called Cyclic Coordinate Descent, first introduced in 1991. It was developed in the robotics field. [5]

The numerical “forward and backward reaching inverse kinematics” algorithm is also a numerical algorithm; it works with constraints. [6] This algorithm in its unconstrained and constraint implementations was used for this thesis. Its unconstrained mean execution time is the lowest, for the current state of art. [7]

For arms, a study on the increase of embodiment has been conducted and compared with a motion tracking solution. The research paper emphasizes an increase of embodiment in virtual reality technologies, by constructing avatars with inverse kinematics generated arms. [8]

Research on Data-Driven models for redundant robot arms has been conducted as well. Problems that Data Driven models have, is the need to be trained first and being only as effective as the training data, they have been fed with. [9]

Hardware sensors and constraints have also been used as an exoskeleton on the fingers, without monitoring each of the segments. The position of not monitored finger segments could be deduced with the right parameters on the constraint model of the fingers. [10]

## Theory

### Planes of Motion

Humanoid Motion can be mapped onto three different planes, these are called sagittal plane, coronal (frontal) plane and transversal plane. With these planes any body part can be modeled with a trajectory.

In a cartesian coordinate system, the sagittal plane is described as the y-z plane. Any movement that goes forward, backward, up or down will display a trajectory with changing values. For example, the head of a walking cycle can be modeled as a sinus curve on this plane, an upward jump will be a straight line upwards and, at the maximum, the trajectory will be a line downwards to a lower spot than initially and then back to the starting point.

The trajectory of a leg for a jumping jack on this plane can be mistaken for a simple forward gait. The coronal plane is clearer for these sideward types of movements. It is span on the x-axis and y-axis, another motion with a trajectory that shows a significant changing rate can be a friendly wave by hand.

The third plane is the transversal plane, or x-z plane, in a cartesian system. Any forward, backward or sideward movement can be detected, a motion that changes the height of an object cannot be detected.

When analyzing a motion at least two planes need to be observed to further classify the motion. Analyzing a motion in three dimensions on a two-dimensional piece of paper requires a lot of visual thinking and even then can still look disarranged.

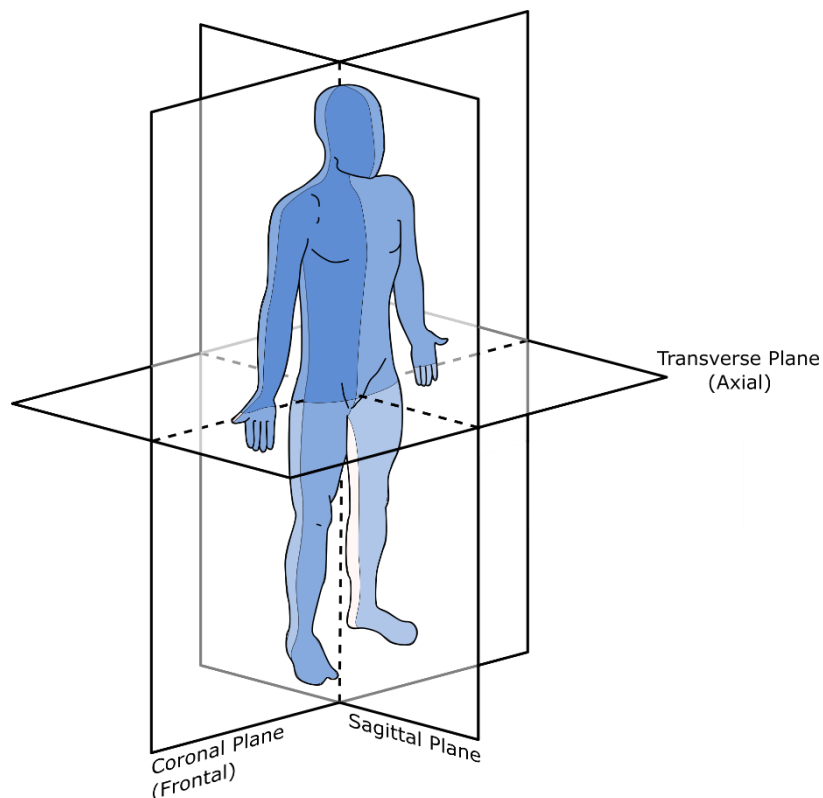


Figure 1: Planes of motion [11]

## Kinematics

Kinematics is the subject of motion without forces in classical mechanics. With the help of linear algebra, rotations, positions, velocities, trajectories, translation, constraints, etc. can be calculated.

Robot Kinematics is a special branch of kinematics, it emphasizes geometry and uses rigid bodies to model physics and motion.

Kinematic chains are links connected by joints. With constraints, motion can be restricted and thus the trajectory specified.

The kinematic structure is a tree or chain of rigid links and joints.

## Forwarded Kinematics

Whenever forwarded kinematics is applied, there is a kinematic chain, where at least the rotation and the length of each segment must be known. With these parameters a position for the end effector, the last joint in a kinematic chain, can be calculated.

## Inverse Kinematics

In the opposite case, where the end position and the length of the links are known, but the angles and positions are not, inverse kinematics will be applied to solve equations for candidates of valid joints positions and angles. There are different approaches to the inverse kinematics problem. An analytical approach sets up equations, that describe the system and then solve for these equations.

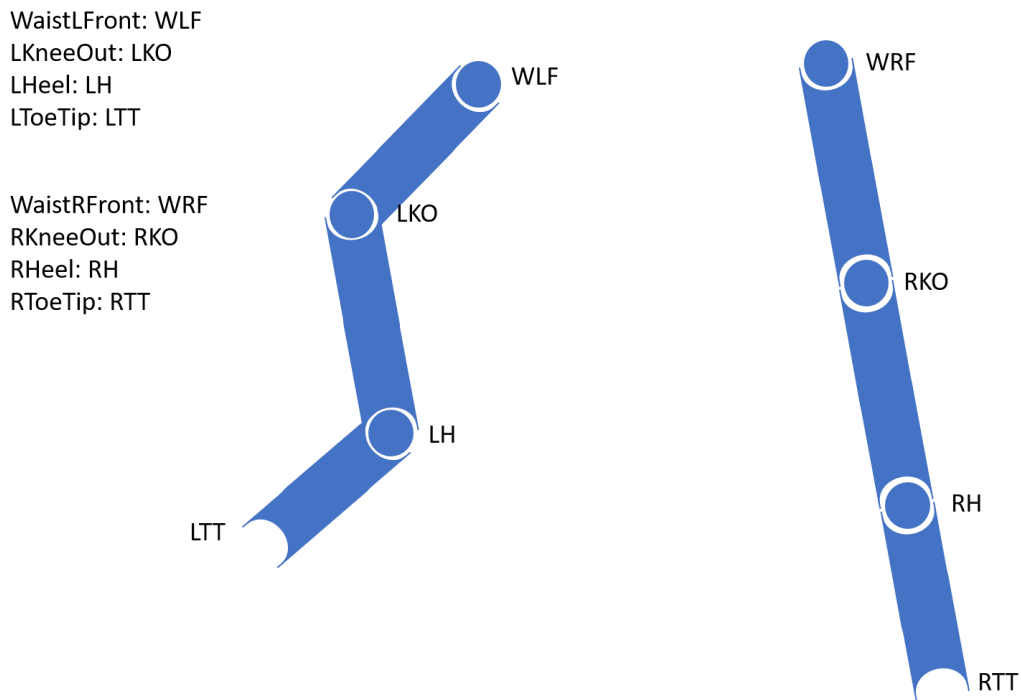
These methods have a high cost, but high precision.

The other methods, numeric solvers approach a problem by a strategy, the Cyclic Coordinate Descent (CCD) Algorithm is one, in this paper the forward and backward reaching inverse kinematics algorithm will be implemented.

## Leg Model

There is a lot of reference as to how a model of the lower body looks like, as we go, kinematic model, it makes sense to model the joints where the knee, heel, waist and toe tip is.

three Variants have been deduced by observing humanoid legs, these will be presented.



*Figure 2: Lower body model with legs separated, unconstrained*

The labeling comes from the motion tracking solution provider, the shortcuts defined in the legend are conventions followed in this thesis.

Figure 2 displays two separate chains modeled as human legs. Each chain consists of 4 joints, whereas the joints that indicate the waist, meaning WLF and WRF are the origins of the kinematic chains. The toetips LTT and RTT are the end of the chain.

As they are separate from each other, a movement that occurs on the WRF joint can't affect WLF and vice versa.

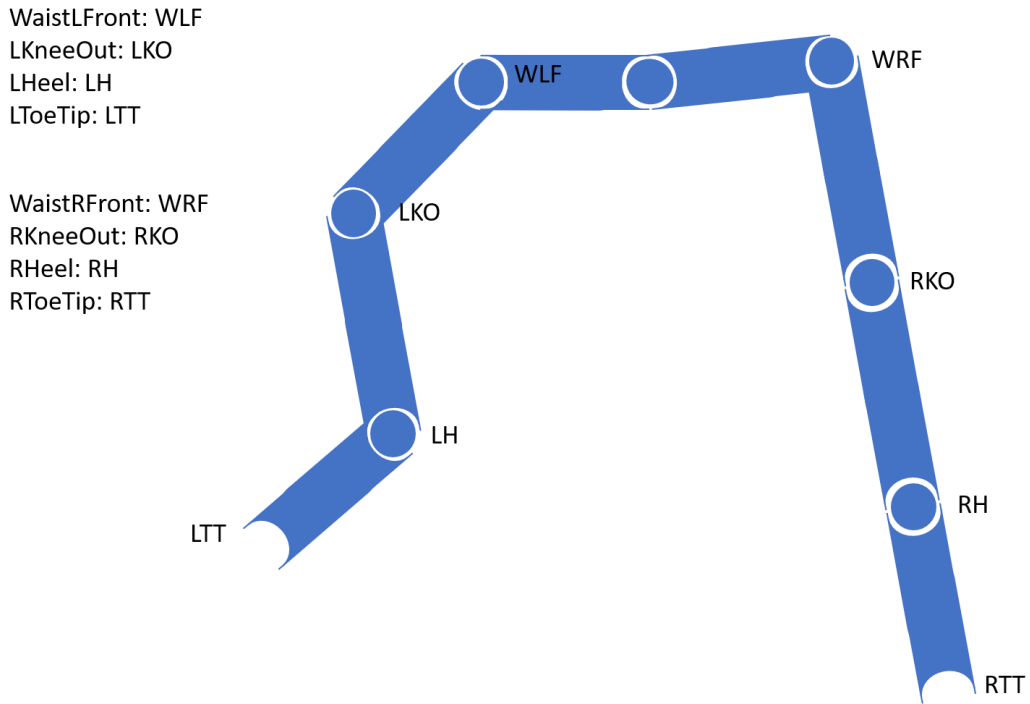


Figure 3: Lower body model, without constraints, with generated sacrum in between

Figure 3 shows a slight variation in the origin of the kinematic chains. For both chains, it is located at the sacrum and is the point in between WLF and WRF.

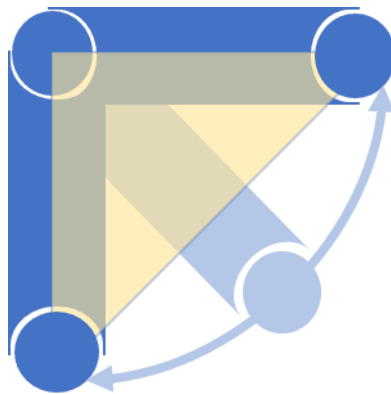


Figure 4: Single link, with constraint

Figure 4 shows a single link between two joints, the yellow triangle is an abstraction of the arc that the joints can move in. In this example the range that the joint can rotate in is  $90^\circ$ . This will affect the position and rotation of the second joint and if another joint is linked to the second joint, its theoretical positions are inside the constraint.



WaistLFront: WLF  
LKneeOut: LKO  
LHeel: LH  
LToeTip: LTT

WaistRFront: WRF  
RKneeOut: RKO  
Reel: RH  
RToeTip: RTT

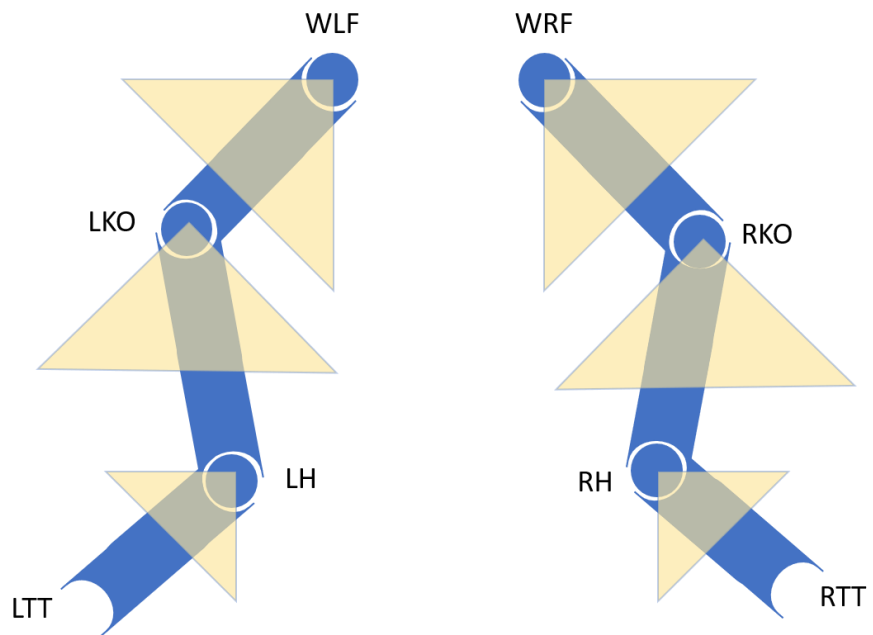


Figure 5: Lower body model for legs separated, with constraints

Figure 5 shows two separate chains with constraints implemented. These are as introduced in Figure 4, displayed as yellow triangles.

The angle constraints can be implemented in different dimensions.

The range of values of the angles of these triangles are one interesting part, as limited joint positions are possible, with this model.

However, this two-dimensionality contains a problem. The triangles must be coils, since the joints rotate the links in at least two dimensions. If one thinks about it, the knee can be moved forward, backward, left and right.

The humanoid skeleton is also limited in its movements. The impossibility of licking one's own elbow is one phenomenon, that is due to the constraints of the elbow joint.

## Fabrik

The Fabrik algorithm by Artisdou and Lasenby is a numeric inverse kinematics algorithm first published in 2011 that avoids the use of matrices and rotational angles. “It converges in few iterations, with low computational cost, and produces visually realistic poses.” [6]

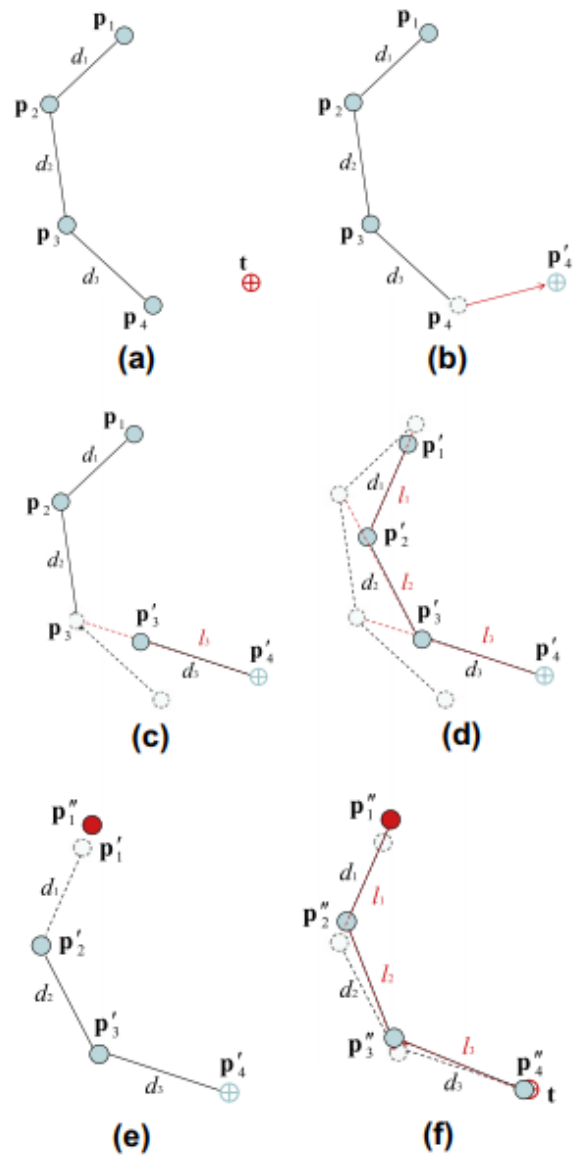


Figure 6: Kinematic chain solved by Fabrik algorithm [6]

Figure 6 shows a kinematic chain, that is solved. One iteration is separated into two phases, that solve the chain from behind to front and from front to behind.

In backward solve the last joint  $p_4$  is set at the target  $t$  position as  $p'_4$ .(b)

Then the joint  $p_{4-1} = p_3$  will be translated into the direction of the set joint in a, meaning  $p'_4$ , with the constraint, that the distance  $d_3$  does not change. (c)

This process is repeated in (d), until the first joint  $p_1$  is translated to  $p'_1$ , with the distance constraint and the direction to  $p'_{i+1}$ .

In (e) the process is reversed, meaning the chain will be solved from the origin to the end-effector. The now translated position  $p'_1$  will be set to the former position  $p_1$ , while being renamed as renaming it to  $p''_1$ .

Lastly, for the iteration in (f), each kinematic joint is translated into the direction of  $p^{i-1}$ , again with a distance constraint. At this step the chain will have moved its position to fit the target slightly and come closer to it.

The algorithm ends, if the target has been reached, with a set tolerance parameter.

## Method

First of all, data needs to be collected and this will be done in the HCI Laboratory of the DMI Faculty of the University of Applied Sciences in Hamburg. The laboratory incorporates a Motion Capture System with 18 cameras and the Motive Tracking Software that creates skeletons, fills out empty marker positions with interpolated values and exports the data. Motion data will be collected and exported in the .c3d data format. [12] [13]

Once this data has been collected, it will be evaluated in the visualization software. This tool takes in the position data in an x-y-z cartesian format, from the c3d file and plots the marker points also into a three-dimensional space. [14]

If the evaluation is satisfactory, the data container will be fed into the tool that will be programmed in the process of this examination. The user of the program needs to specify the points form a kinematic chain for the left and for the right leg. Afterwards, the solver will start calculating positions for every frame of the motion format.

After a successful run of this process, a new file will be written from a user-specified file directory, with a .csv file that contains both acquisitions to, therefore, calculate the difference in distance for the method used. They are sometimes referred as delta values in this thesis. Also, distances.csv will be output that contains the distance values between each joint, for each frame.

In the visualization software, the output motion format can be observed to see an immediate result. Furthermore, the calculated points will be compared to their coordinate values in more detail. A comparison can be performed either in two different tabs of the software or with a flag entered in the self-written solving program.

## Motion Tracking

### Process

One can describe the markers as small round spheres, that are white and reflective. As this marker position deduction is part of an image processing, noise can also occur. This noise can be reduced by removing white or reflecting objects from the tracking space. If these objects are

part of the tracking scene, they can also be masked with a non-glossy tape or masked with a tool in the tracking software.

Calibrating the cameras for the tracking space can often be required whenever camera positions are moved. Symptoms of a noncalibrated tracking space are jumping coordinates from one point to another with any motion performed.

The calibration is performed with a wand stick, which has 3 markers located at the end, with a gap between the marker neighbors. This calibration process requires the operator to make samples inside the tracking space, by waving the wand stick in circulation motions. The goal is to have a high number of samples, for high quality in the precision of the measurements of the tracking space.

It is a process that can take up to 20-30 minutes, to get 10.000 samples for each camera. Depending on the relationship of the camera views to each other, the rate of samples per camera can increase faster or slower. In general, it can be observed, that if the viewport of a camera overlaps with other viewports of other cameras, the increasing rate of samples is higher. This can be a criterion on where to position the cameras in the setup.

After the calibration, the cameras will be localized in relation to each other in the coordinate system of the Motive application. At this step the ground must be defined with the calibration triangle. It also has three markers attached to it and it is formed like an “L” .

After a successful calibration process, the trackers can be arranged on the motion tracking suit. The components of the suit are a zipper, pants, flats and a cap, all made from the same material. Its material has a Velcro®-friendly surface as the tracker is fixated with Velcro®.

Once the markers are in position for the default OptiTrack arrangements, a skeleton can be generated, which enables export for motion file formats with a hierarchy encoded.

These are the technical steps required for the complete preparation of a motion capture session.

#### Design of the Motion Set

For this thesis, a set of motions has been specified, before any motion tracking session. For the first capturing session, no special category has been selected. They can be categorized according to the planes of motion.

|   | <b>Sagittal Plane</b> | <b>Coronal Plane</b>  | <b>Traversal Plane</b> |
|---|-----------------------|-----------------------|------------------------|
| <b>Jumping Jack</b>                           | X                     | Oscillating on x-axis | Oscillating on x-axis  |
| <b>Forward-Backward Sagittal Jumping Jack</b> | Oscillating on z-axis | X                     | Oscillating on z-axis  |

|   |                          |                       |                               |
|---|--------------------------|-----------------------|-------------------------------|
| <b>Cross jumping jack left leg front</b>  | Oscillating on z-axis    | Oscillating on x-axis | Oscillating diagonally on x-z |
| <b>Cross jumping jack right leg front</b> | Oscillating on z-axis    | Oscillating on x-axis | Oscillating diagonally on x-z |
| <b>Walk on the treadmill at 3 km/h</b>    | Oscillating on y-axis    | Oscillating on y-axis | Oscillating on z-axis         |
| <b>Walk on the treadmill at 5 km/h</b>    | Oscillating on y-z-plane | Oscillating on y-axis | Oscillating on z-axis         |
| <b>Jog on the treadmill at 7 km/h</b>     | Oscillating on yz-axis   | Oscillating on y-axis | Oscillating on z-axis         |

*Table 1: Categorization of knee joint movement on planes of motion*

Table 1 shows the set of motions and the expected movement of the knee joint on the planes.

The cell that is described by the sagittal plane column and the jumping jack row contains an X, which means it is expected that the position will stand still in comparison to the other planes in the model. On the coronal plane, an oscillation on the x-axis will be expected. On the traversal plane, the oscillation on X should also be visible.

Cross jumping jacks show oscillation in two directions, so on each of the planes, an oscillation is visible. One of the planes will also show an oscillation in two directions at the same time. It will either oscillate diagonally or in a circle.

As one can see, whenever a motion in one direction occurs, movement of a joint is always visible on two planes, due to the nature of three-dimensional space.

#### Motion Tracking Skeleton Model

For the beginning, the default OptiTrack marker setup for legs will be used. It consists of 20 markers, ten on each side of the lower body.

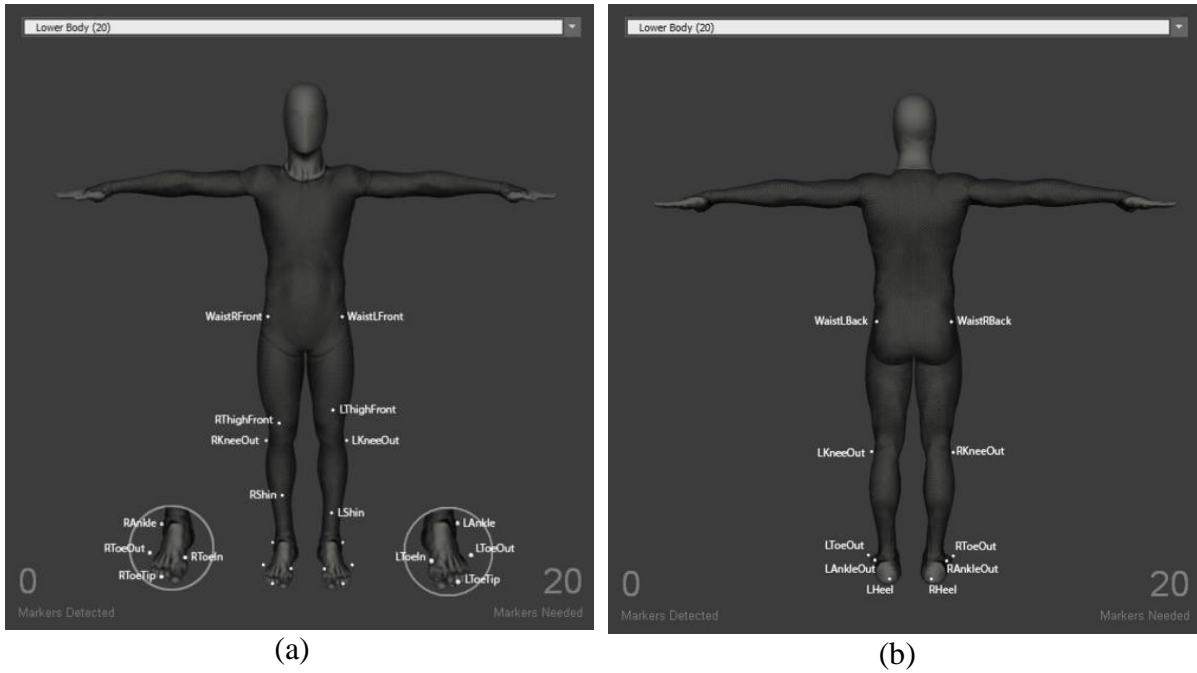


Figure 7: Baseline Lower Body marker-set

Figure 7 depicts a coronal of the human body from (a) the front and (b) the back. This is one of the marker-sets that have been used for the data collection. For the models, the relevant markers are knee, heel, waist-front and toe tips. The toe-tip joint is the end-effector and the waist-front is either the origin for the unconnected unconstrained and unconnected constraint model and for the connected unconstrained model it is the second joint in the chain.

Another marker-set, that has been used to collect data is the Rizzoli Lower Body Protocol, displayed in Figure 8. Originally for this protocol 26 Markers are used, the version integrated into Motive also contains the RSK and the LSK markers as visible in Figure 8, that is 28 markers in all, to better “distinguish the left and right of the skeleton”. [15]

Six of the markers in this protocol are used as calibration markers to “represent internal anatomical landmarks in the largest possible number in order to limit the necessary marker-set”. After the calibration process, they should be removed from the motion tracking suit, as they are not a requirement for the tracking process, only for the generation of the skeleton. [16]

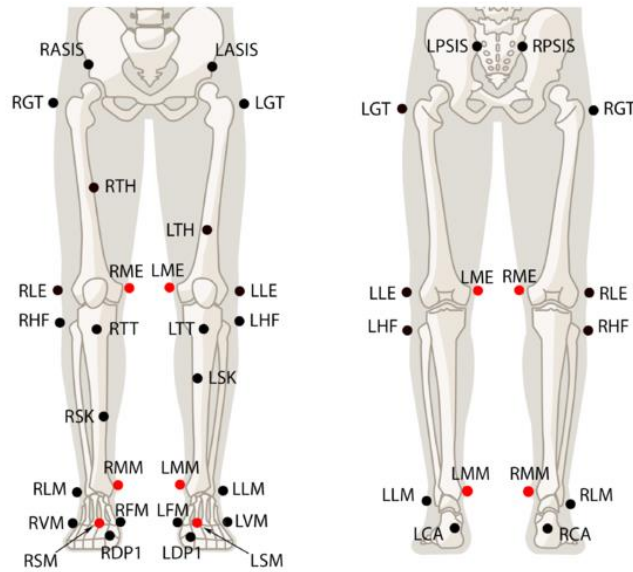


Figure 8: Rizzoli Lower Body Protocol

For the implementing of the model, on the right leg the 4 markers RGT, RLE, RCA, and RFM are used. Optionally when 5 segments are used, the RASIS marker will replace the origin. The marker-set on the left leg is symmetrical to the right one, LGT, LLE, LCA, LFM, and LASIS. Additional descriptions of the short form can be found in the Appendix

#### Data Export

Once all the tracking data has been captured, it needs to be exported in order to feed it in the solving program. For this process there are different options. Motive supports several formats like .C3D, .bvh, .fbx, .csv, .trc. All of them have their use cases, but for the beginning, an emphasis has been put on the .c3d format has been put. It is enough for our use case, hierarchical structures are not required and read- and writability speed and size is optimized, due to its binary format. As one sees in the comparison between Figure 9 and Figure 10 the file size of the format saved in .bvh is more than double the amount of the file saved in .c3d. After removing the hierarchical structure of the .bvh file, it is 4kb smaller, so not significant for this comparison. Additionally, the c3d format is public domain.

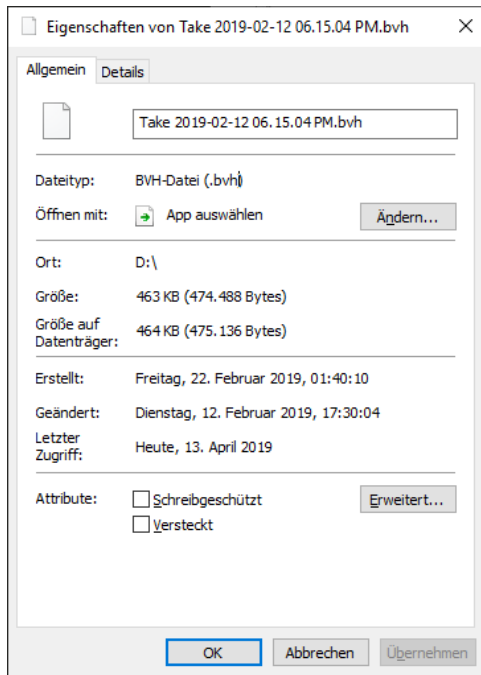


Figure 9: File size of the bvh format

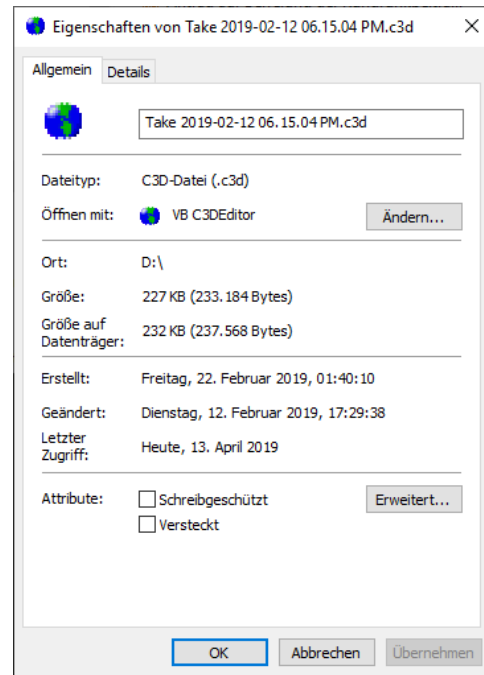


Figure 10: File size of the c3d format

## Infrastructure and Solving Application

Initially, there were technical constraints involved in developing the application. Execution speed is important, since implementation of the program as part of a pipeline needs to be time-cost efficient to render frames in real time for a satisfying result, thus the object-oriented programming language C++ has been chosen as a development platform. One library involved in the development of the application involves code older than 5 years, with source code that is not up to C++11 standard. This library could not be used on a modern system, like Windows 10 or Ubuntu 18.04.

To overcome this problem the application was developed in a virtual box and the setup processes automated with Vagrant software. Later in development it became clear, that writing a file into a synced folder<sup>1</sup> from the program is not possible. Write permission is given for the program, but for the synced folder, in which the program build and the source code are contained, files could not be written into these types of folders. It is possible to write files into a directory, that is not a shared folder and copy them into the shared folder. This workaround has been adopted for the development, testing, and usage of the application.

As to executing the program, Figure 11 displays a flowchart of the application.

<sup>1</sup> A synced folder in Vagrant is a folder, that synchronizes every file in both environments, meaning that when a file is being copied, moved or deleted in the host environment, the same operation applies in the virtual environment and vice versa. [21]



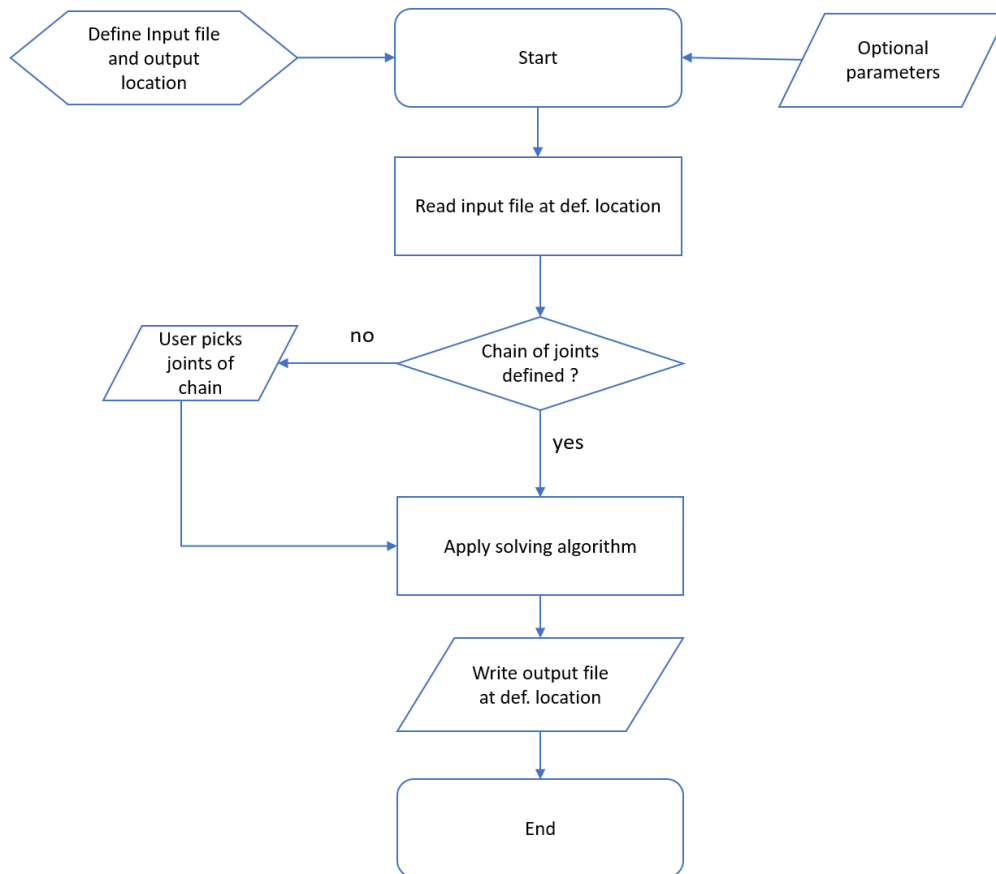


Figure 11: Flowchart of the program

Before executing the program, an input and output file must be specified, that need to be in the .c3d format. With this format a basic run is possible. The additional parameters specify the variant of the inverse kinematics algorithm, that will be executed. Optional parameters can also be inserted into the command line interface. (See Appendix A for the specification)

The command line interface will be executed in the format displayed below :

```
./IKAlgorithms <c3dinputfile> <c3doutputfile> [optional parameters][optional preselection]
```

Interface 1: Usage of the application written

Whereas ./IKAlgorithms is the name of the program, <c3dinputfile> is the placeholder for the file, that contains raw point position data for the calculation and <c3doutputfile> is the placeholder for the file with the modified point data, that is being calculated with the inverse kinematics algorithm.

The optional parameters enable the user to execute the program in different variations, and different algorithms other than the default. So far the variations “unconstrained”, “unconstrained with static interpolation between first joints” and “constrained” are possible.

When the program is being executed, the first process will be to read the c3d file. This functionality is implemented in the biomechanical toolkit by an “AcquisitionFileReader”. Its filename is set, the readers are updated and the output is returned, as an instance of the Acquisition. The complimentary writing process is implemented in the library in the form of the “AcquisitionFileWriter” class.

From an Acquisition class object, point labels, points and ultimately the position of the points per frames can be deduced. [17] This should occur once the ids for the points are known.

The optional preselection will reduce the overall execution program, as the step of the input, that the user is required to enter, is already served at this step, thus skipping the step of defining the joints one by one in the execution of the program. Samples of this method can be found in the samples.txt file, included in the project folder of the application.

After this process the actual calculation will begin, the forward and backward reaching inverse kinematic algorithms will solve the chain for satisfactory candidates.

Integrating the position of the estimation into this process will override the position data of the point on each specified frame. The label of the points that are specified as joints of the chain will have an addition “fabrik”-tag in their label.

Additionally CSV result data will be output in the result folder, which contains the distances between each joint and a delta file.

The distances will give additional certainty about the correct execution of the algorithm. The delta file contains position data of the measurement .c3d file, the modified .c3d file and the deviation of measurement position and solved position.

## Mokka Motion Analyzer

The Mokka Motion Analyzer developed by Arnaud Barré will serve as a visualization platform. It can read a variety of motion tracking formats including the formats Motive is capable of exporting, .trc and .c3d.

The function of drawing trajectories and color coding for the markers is used to visualize the movements of the solved points. The functions used in this thesis are displayed in Figure 12.

Important note: y is the forward axis and z the vertical axis in the coordinate system

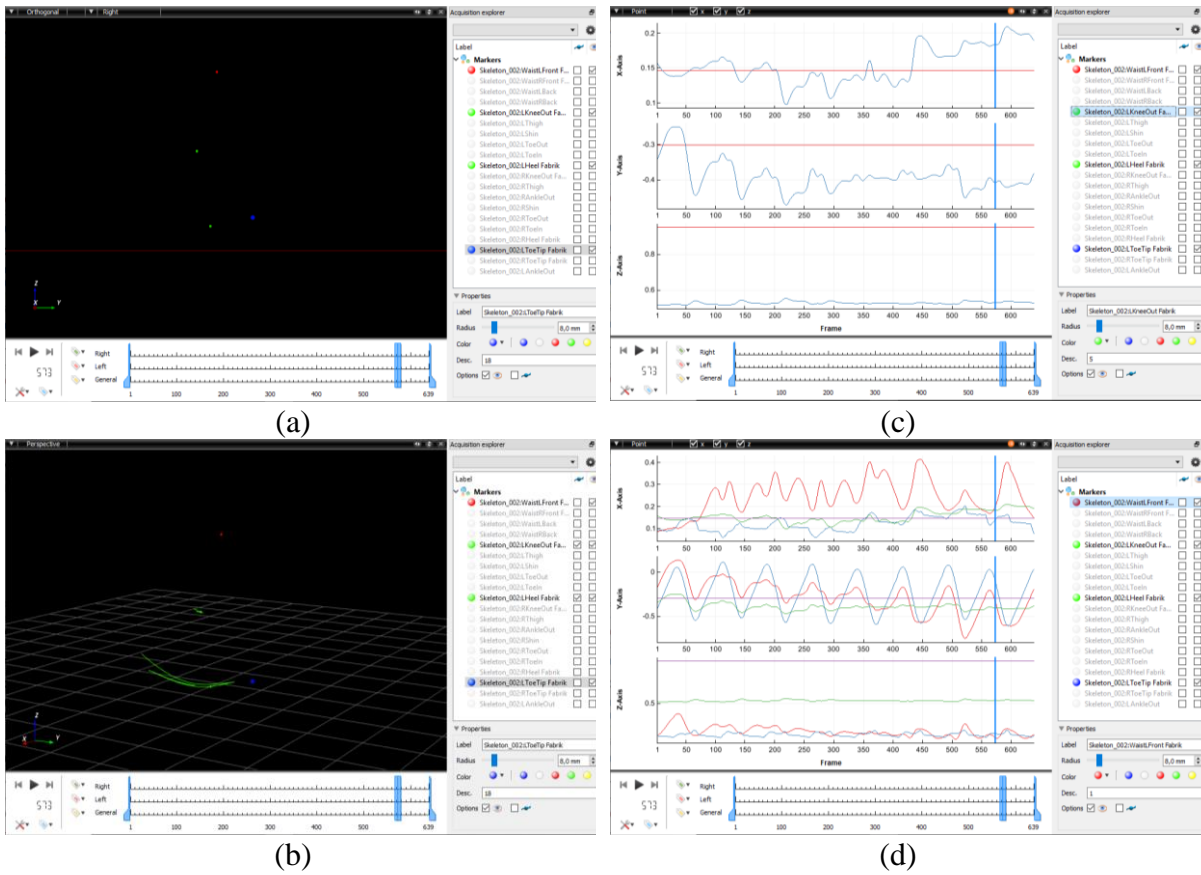


Figure 12: Visualisation of the result/unconstrained/jog\_001.c3d dataset in Mokka, only the markers WaistFrontLeft, LKneeOut, LHeel and LToeTip are enabled

(a) Orthogonal view from the right on the markers from the on the enabled markers with color coding of the markers, (b) Perspective View on the Markers with trajectory setting on the green markers, (c) Position coordinates of Waist Front L (red line) and LKnee (blue line), (d) Position coordinates of Waist Front Left (red line), LKneeOut (blue line), LHeel (green line) and LToeTip (violet line)

## Problem and Solutions

During the development of the method some minor problems have been occurred, that might deserve to be documented, for not losing to much time, when they are reoccurring.

One of them involves using different versions of the motion tracking camera in one setup.

Optitrack cameras must be wired in a specific way, to work together. For that, they have an output- and input-signal and an USB-connection to an Optihub. One Optihub has six USB-slots for Optitrack cameras. If one wants to use more than six cameras, more Optihubs must be used, which then again also must be wired with each other, with a synchronization cable.

In the end one of the hubs must be connected with the computer, where the Motive software runs, then the number displayed on the camera display is mapped to the designated camera view of the software.

For some reason, when an old cameras number in the sequence is after a new camera it start the number sequence from one again.

To get the right enumeration the following method has been done.

Six of the old cameras have been plugged into the first hub and six of the new cameras have been plugged into the last hub. Since neither six old nor six new cameras are in stock, but 5 old cameras and one new, they must be mixed for the port in between.

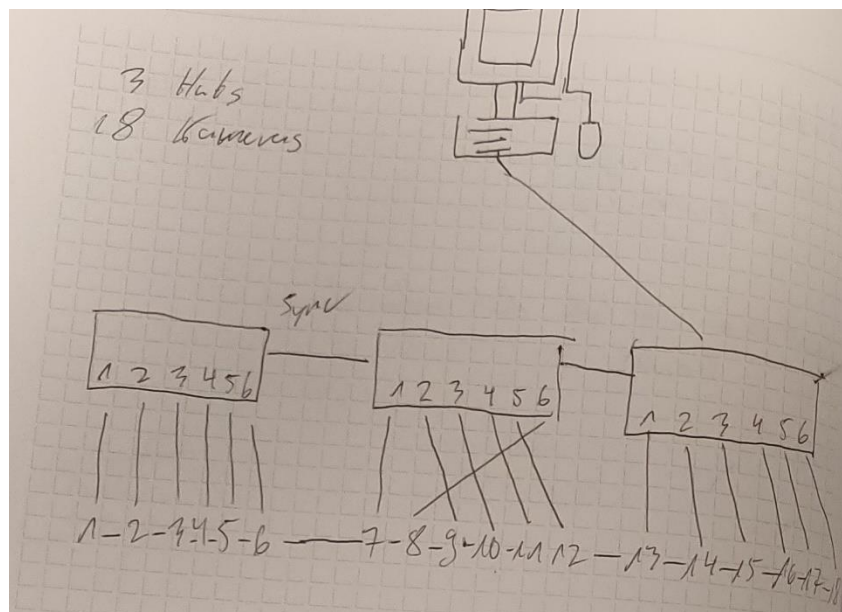


Figure 13: Eighteen Cameras and three hubs, Motion tracking setup

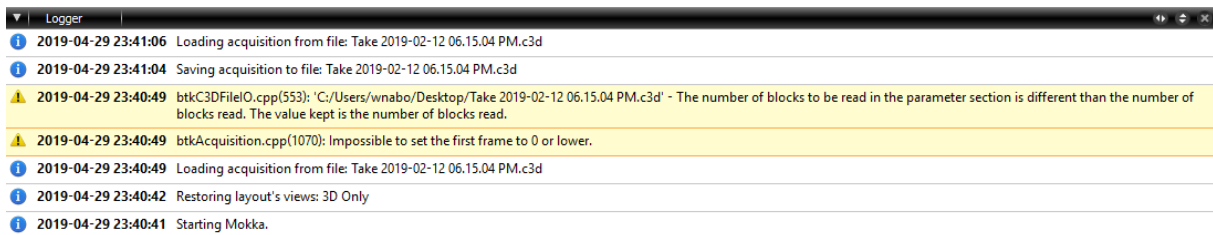
There has been several trial and error involved but in the end the solution visible in Figure 13 has been found. In that case the camera 7 is of version V100R2, so a new camera and cameras No. 8 to 12 are version V100 Cameras, so old ones. The trick is to plug in camera 8, so the camera with 7 and 9 as neighbors, into the Optihub slot 6 and not 2. That is quite a bit

unintuitive, however the mixed use of different versions of cameras in one Optihub is not designated. The rest of the cameras and their neighbors are wired, the way one would expect them to but with an offset of -1. So, 9 into the slot for 2, 10 into the slot of 3 and so on.

Another problem involved the exported c3d data of the Optitrack measurement and the use of the c3d file reader of btk.

The c3d data exported by optitrack involves data at the frame zero. Opening a c3d file with a btk c3d reader will result in a read error.

This problem has been solve using the Mokka visualization software. Interface 2 shows a log of the procedure, to acquire a functional file for the btk framework.



*Interface 2: Log of an when opening, saving and reloading an c3d export Optitrack style*

This problem can be resolved by opening the c3d file in Mokka first, then two warnings will be logged. First that setting first frame to 0 or lower is impossible. Second that program noticed the file saved on the hard drive and the acquisition read into the working memory show differences in the number of blocks. The file will however still be opened, without frame 0, in the acquisition.

Saving the acquisition at this point will save a file, without the frame 0, therefore loading it new will result in no warning and consequently no error, when used with btk.

## Result

### Visual Analysis

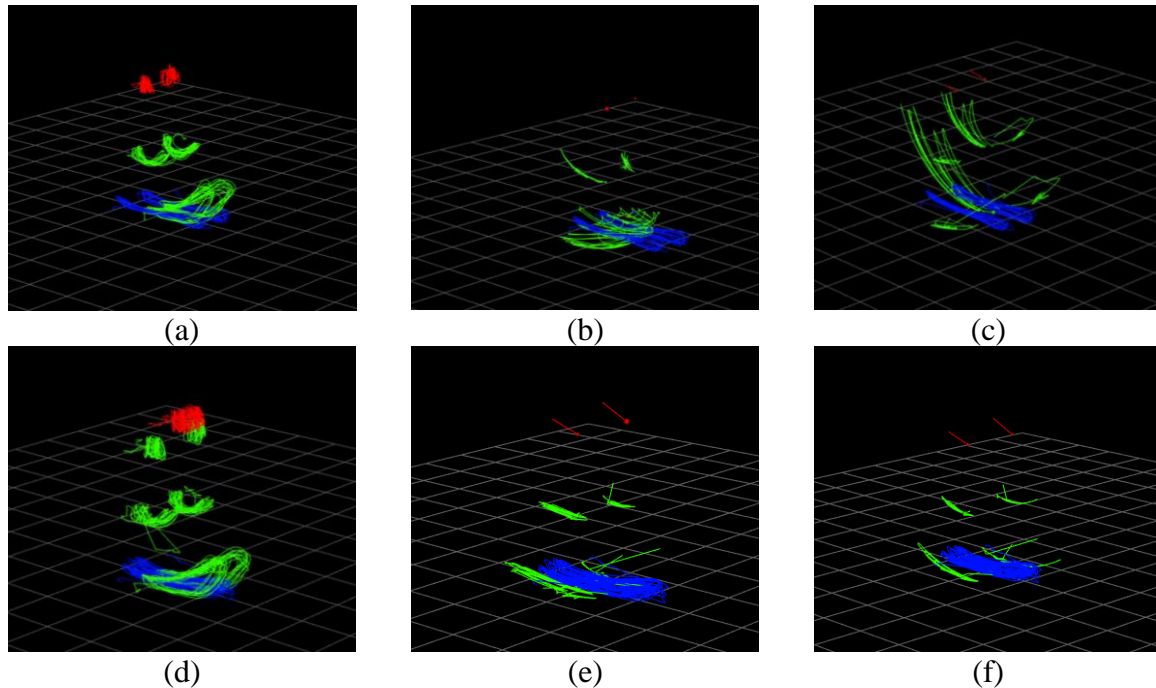


Figure 14: Trajectories of the joints for a jogging motion with a lower body baseline marker-set for (a),(b),(c) and for Rizzoli lower body protocol (d),(e),(f). (a) Joint trajectories for measurement; (b) Trajectories for Fabrik unconstrained; (c) Trajectories for Fabrik constrained; (d) Joint trajectories for Rizzoli marker-set measurement; (e) Trajectories for Fabrik constrained without horizontal angle constraint block; (f) Trajectories for Fabrik constrained with horizontal angle constraint block

Visualization of one sample of the data is visible in Figure 14.

Since the jogging motion is periodic, parts of the trajectory remain on the same trail. This is most noticeable in the measurement data (a) and (d). The targets in blue also show this characteristic in every image, as they are bound to the model and taken from the measurement.

The green trajectories in (b), (c) and (f) can be described with geometric mean as an arc, whereas the knee marker of (c) indicates the longest angle of the arc, compared with the other three visuals. The heel trajectory of (f) can also be described as an arc, with its central point further away from the waist, towards where a torso of a human body would be. Visually the arc length would be shorter.

The arc of the trajectories for the markers of the right and left leg are parallel to each other for (c) and (f), whereas the arc trajectories for the arcs on (b) are not. The heel-marker on (b) scatters into many different directions, the trajectories for the left and right heel overlap as well.

The trajectory in (e) does not resemble an arc. For the knee markers it can be described as a translation forwards and backward. The heel markers behave similar, even though one of the markers is not in sight.

The origin position in red, for baseline the Waist Markers (a), for Rizzoli the axes (d) is not completely static at one point in the measurement, it displays scattered movement around one point. While for the unconstrained (b) trajectory the origin point stays in one position. On the constrained (c), (e) and (f) the origin moves slightly on a path, with visually marginal to no amount of movement.

As to the interpretation of the visuals, one can say that visually the (f) looks closest to the measurement data. The length of the arc is close to the size of (d) and the leg trajectory is parallel to each other. However, one also must note that the trajectory is not very “dense”, meaning it would not move much.

Another aspect is that the segment from heel joint to tiptoe joint varies in length, it changes in general for the data-set for the kinematic chains as one can see in Figure 15.

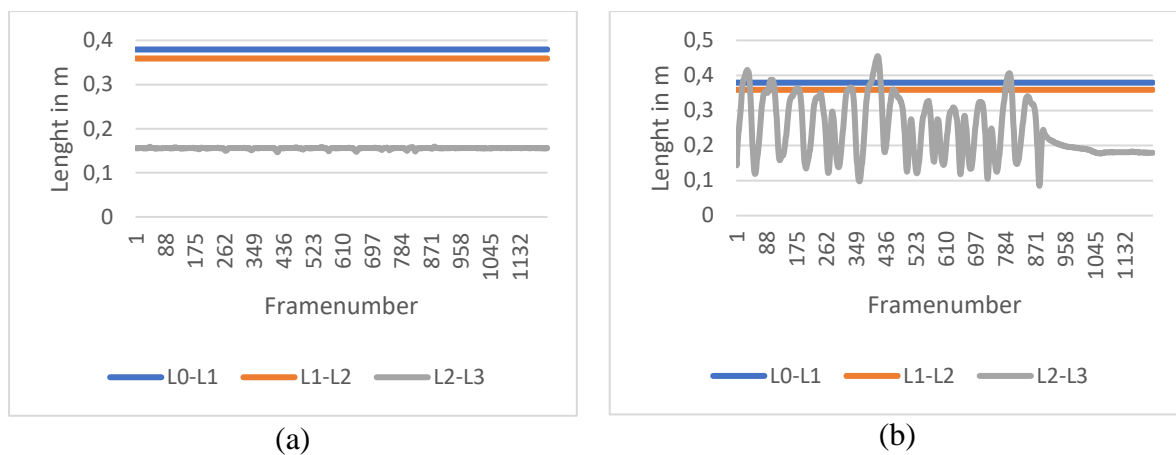


Figure 15: Segments lengths for (a) unconstrained kinematic chain and (b) constrained kinematic chain for a jogging motion with the Rizzoli lower body protocol. The joint mapping for this data-set:  $L0 = LGT$ ,  $L1 = LLE$ ,  $L2 = LLM$ ,  $L3 = LFM$

As one can see, the distance stays constant for the segments of the unconstrained chain, whereas the distance for the last segment of the constrained kinematic chain scatters from less than 8 cm to 42 cm.

### Jogging Motion in Different Variants

The positions in the measurement of the jogging motion have also been compared to the different variants of the algorithm. Subtracting the position vectors from the captured data and the Fabrik calculated data and then calculating the absolute of the result will return a length in meters. This length, it is referred in this thesis as delta, is an indicator for the quality of the algorithm. The bigger the delta, the more the calculated position is off from the tracked position.

If delta is 0, the model can reconstruct the measurement. If it is 0,5 it is 50 cm away from the measured position.

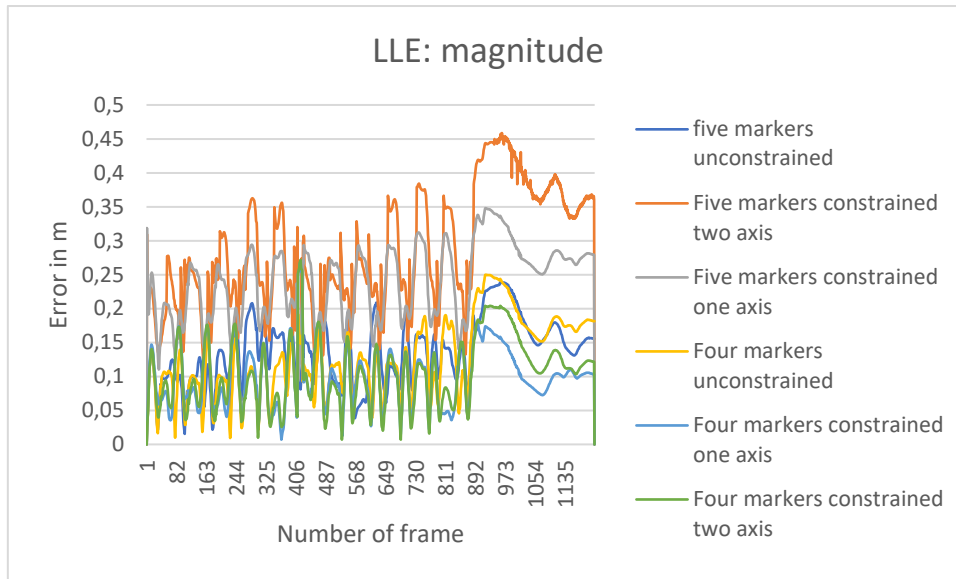


Figure 16: Delta value for jogging motion in different variant, with the Rizzoli protocol

Figure 16 depicts delta vector magnitude graphs for different models the algorithm can be applied to a data-set for the marker LLE of the Rizzoli protocol. For each frame, a position has been captured. The frame number tells at which point in time the measurement has been taken place.

The vertical axis displays the error of the algorithm applied to the data-set, the delta value.

The first observation about the error behavior of each graph and the amplitude of the oscillation is not constant, it changes each period. The jogging motion is also periodic until frame 892 is the frame when the treadmill is left, for each graph.

At first glance, it seems as if each graph just has an offset from another, however on second look, amplitudes can also differentiate. The graph for four markers constrained two axes and the graph for five markers constrained two axes. The constraints for this data-set are as follows.

The error on four constraints is in general lower than the one in five constraints. This can be seen by the mean. Whereas for constraints mean is approximately 10,6 cm, the LLE, the mean for five markers approximately 35 cm.

|       |   |
|-------|---|
| LPSIS | Up: 10.0, Down: 10.0, Left: 10.0, Right: 10.0 |
| LGT   | Up: 10.0, Down: 10.0, Left: 10.0, Right: 10.0 |
| LLE   | Up: 30.0, Down: 30.0, Left: 1.0, Right: 1.0   |
| LLM   | Up: 10.0, Down: 10.0, Left: 10.0, Right: 10.0 |
| LFM   | Up: 0.0, Down: 0.0, Left: 0.0, Right: 0.0     |

(a)



|     |   |
|-----|---|
| LGT | Up: 10.0, Down: 10.0, Left: 10.0, Right: 10.0 |
| LLE | Up: 30.0, Down: 30.0, Left: 1.0, Right: 1.0   |
| LLM | Up: 10.0, Down: 10.0, Left: 10.0, Right: 10.0 |
| LFM | Up: 0.0, Down: 0.0, Left: 0.0, Right: 0.0     |

(b)

Table 2: (a) Constraints for a kinematic chain of 5 elements, Rizzoli protocol, left leg, with two axes used, (b) Constraints for a kinematic chain of 5 elements, Rizzoli protocol, left leg, with two axes used

The error maximal error peak reaches their heights from 900 to 980 and for these two graphs five constraints, two axes are also higher than, four constraints two axes. So, the error is maximal when the treadmill is left.

It is also notable, that in Appendix B, where the position is on its components, the abs x shows the highest error

So, the result is dependent on the constraints that are set for the joints.

## Trajectory Error of Motions<sup>2</sup>

For the next part, two classes of motions will be compared with each other. Forward movement was measured in different sorts of speed, at 3 km/h (Walk slowly), 5 km/h (Walk faster) and 7 km/h (Jog). The other cases that will be compared with each other are jumping jacks into forwarding direction, jumping jacks for a sideward direction and cross jumping jacks. The model variant is not varied, it stays consistent in these comparisons.

Case: Treadmill Data-set, Constrained, Knee Joint

The movement data-set that is captured for longest of the three is the walk faster 001 data-set, it takes 881 frames, then jog 003 with 881 frames, then walk slow with 821 frames. For the constraint model the ranges of error the show quite similar characteristics. For knee, just as for the heel, the error graph changes in cycles. This is can be seen for all three graphs.

As to for the x-error of the knee one can say, that the walk faster motion by mean shows the least error of 16.38 cm. Once the first fifty frames have passed, the error on the x-axis ranges from 7 cm to 13 cm for the walk faster data-set.

---

<sup>2</sup> Note: For the diagrams only a sample of the diagrams are inside this main part, all of the diagrams can be found in the Appendix C: Measurement Tables

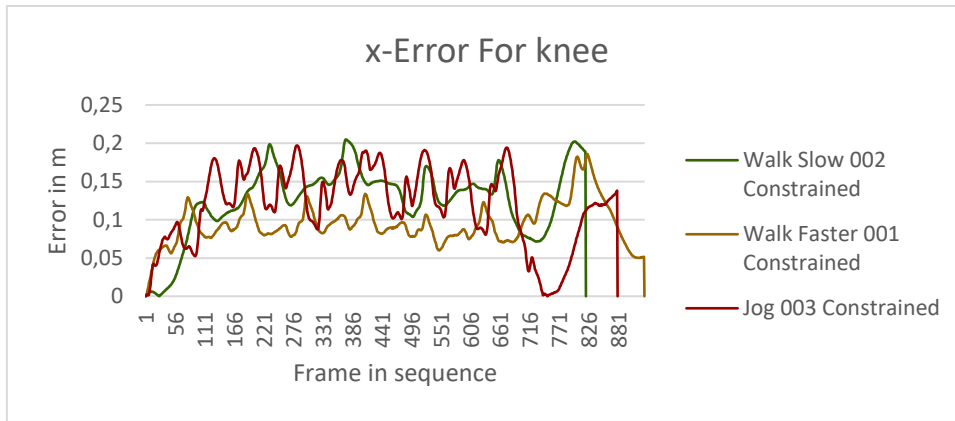


Figure 22: Error behavior for the left knee of the treadmill data-set constrained, absolute x-error

For the walk slow data-set, the range for the error on the x-component starts at frame 110 to rise fast and the error ranges between 4,8 cm to 19 cm. Lastly, the jogging motion error starts at frame 111 to behave periodically. The jogging motion ends at frame 881, the walk faster motion at, the walk slow at 821.

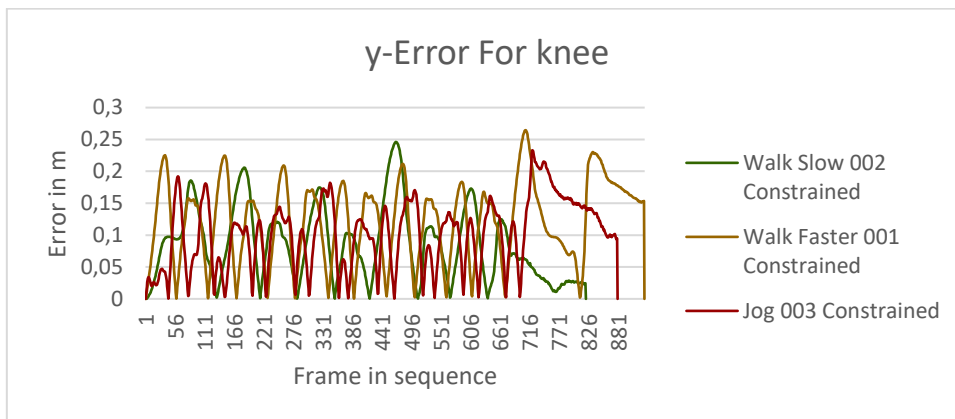


Figure 22: Error behavior for the left knee of the treadmill data-set constrained, magnitude error, absolute y error

For the y-error of the knee, the graph behavior is difficult to evaluate, the graphs error all ranges between 0 and approx. 25 cm. Walk faster reaching this peak close to frame 716.

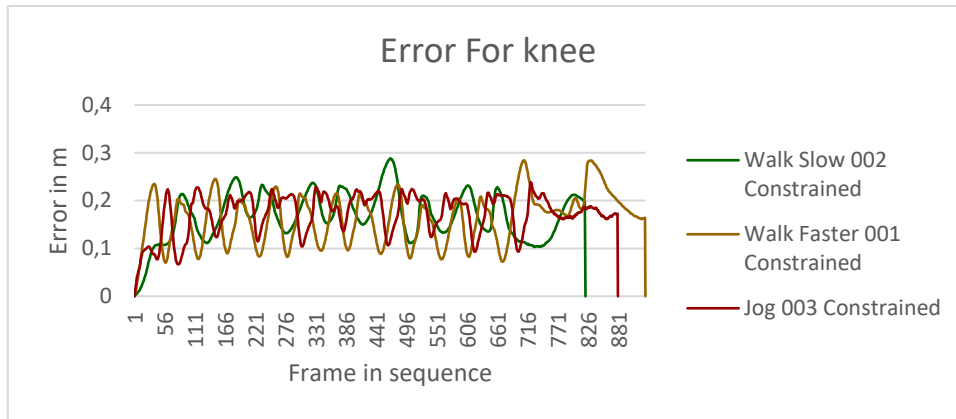


Figure 22: Error behavior for the left knee of the treadmill data-set constrained, magnitude error

The general error for the knee for a jogging motion for the knee is oscillation between frame 24 and 770. For the walk slow motion, it reaches approximately 27,59 cm as maximal value. The oscillation range of the motion is mostly between 9 cm and 21 cm for the walk faster graph especially, the jog is also moving in between these values, however, the oscillation is quite more irregular. The oscillation for walk slow is taking place in between 11 cm and 22 cm.

When looking at the components of the knee error, it is visible, that some components don't share as much of the amount of the magnitude error, than others.

By comparing the mean value of the error components of the jogging measurements, which is 17,24 cm, the x-error shares the biggest amount with 11,93 cm of the error, while the y-error mean is 10,10 cm and the z-error mean 3,15 cm. One must note, that the accumulated amount of the components can be bigger than the mean error of the magnitude, as for the component error the absolute value has been calculated.

#### Case: Treadmill Data-set, Constrained, Heel Joint

For the heel the behavior looks quite similar on the first view, x-error and y-error behave similar, oscillation for the same sequence of frames, y-error very with a high frequency of oscillations for all the graphs.

On the second view, however one must notice the scale. The oscillations occur in between the range 1 cm and 33 cm for x-error, the graphs of the y-error reach peaks of 40 cm to 45 cm. The jogging motion graph on the z-error does not oscillate as frequently as the jogging graph of z-error on the knee. The scale for this error is also higher. The z-error graph for jog reaches peaks of 30 cm, while the jog-graph on the z-error for knee only reaches 9 cm. This can also be seen in the mean values, which are 3,15 cm for the knee on the jog graph and for the heel, it is 9,5 cm.

The general mean error on the heel is 35,46 cm, the x-error mean is 24 cm, 20 cm on the y-error and 9,5 cm on the z-error, as mentioned.

This makes shows that the heel-error is, in general, higher for heel a jogging motion than the knee, for a kinematic chain with constraints calculated with the jogging motion.

Case: Jumping Jacks Data-set, Constrained, Knee Joint

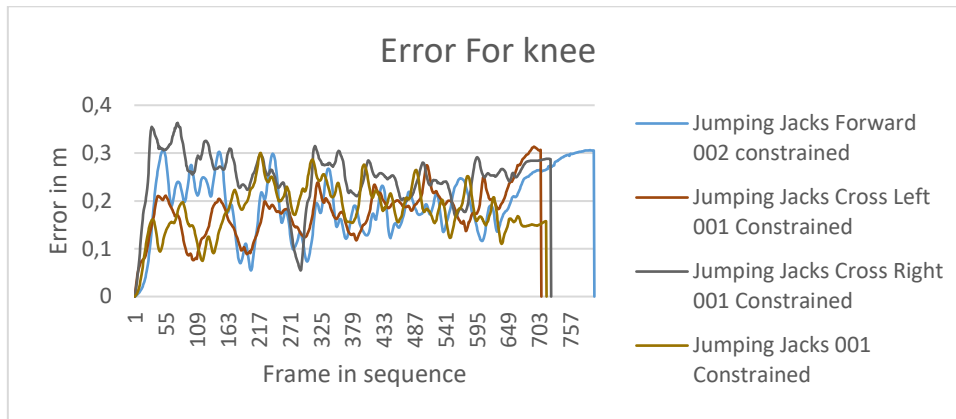


Figure 24: Error behavior for the left knee of the jumping jack data-set constrained

More sets of motions, that will be compared are the jumping jacks set. Here also the absolute x-,y-,z-error and magnitude the error-vector on the knee and heel in meters, has been plotted as a graph per frame. The kinematic chains are all constrained in these sets.

The graphs are plotted for 757 frames, as the jumping jacks forward set is this time long. The other graphs end close to frame 710.

The x-error for the knee on the jumping jacks is oscillating, this is visible for the jumping jacks data-set. It the amplitude of the oscillations is around 12 cm long, a trend is not visible.

A positive trend is also visible for the jumping jacks forward, the maximal error here occurs at the last frame. It is assumed that the trend occurs due execution of the motion becoming slightly more horizontal.

The cross graphs for the x-error also trend slightly positive, cross right more than cross left. The oscillation is not clearly visible as in the jumping jacks.

For the y-error of the knee, the trend and oscillation behavior will also be observed.

For the trend behavior of y-error number for jumping jacks forward are trending negative and range from 0,3 to 0. At frame 649 it is increasing in error to 29 cm at the end of the motion.

The jumping jacks forward graph error behavior trend is decreasing and oscillating to frame 703. The oscillations of most of the graphs occur in the range of 0 cm to 27 cm, except for the jumping jacks cross left, which have one maximum close to the end of the motion, at frame 690. This graph is trending positively.

Jumping jacks cross right on the other hand is trending negatively and oscillating not as clearly as the other data-set.

For the z-error behavior, the graphs are oscillating highly frequent between the values 0 and 10 cm, from the start of the motion at frame 1 to the end of the frame. For the jumping jacks-

constrained graph the error oscillates from 0 to 1 every 20 frames, which is quite noticeable, as all the graphs of the cross-jumping jacks, jumping jacks and forward jumping jacks, are oscillating at this frequency and amplitude.

The error in magnitude of the vector shows a range of 0 to 35 cm. Trends seem little to no noticeable, however, the graphs can be described with more than one oscillation per graph and with different wavelengths.

The jumping jacks forward graph, for example, can be described with one period taking 20 frames another period taking place every 110 frames.

The jumping jacks cross right error magnitude oscillation can also be described in with a slightly negative trend and an irregular wavelength, that ranges between.

The jumping jacks cross left error magnitude oscillation, on the other hand, is trending positively, with the first oscillation wavelength being around 150 frames long and the second being around 10-20 frames long.

Lastly, the jumping jacks graph behavior oscillates without a trend, also with two overlapping oscillations, one with a longer wavelength of 170 frames and a shorter of around 20 frames.

The mean of the jumping jacks forward being 19,45 cm error per frame long, on the y-axis component being more prevalent, with a mean of 14 cm error per frame. The x-component having a mean error of 9,7 cm and the z-component 4,0 cm.

For the cross-jumping jacks left the mean error magnitude is 17,75 cm long, with the x-component having an absolute error of 13,75 cm, the y-component absolute 7,8 cm, and the z-axis 5,1 cm. So, the x-component having the biggest amount of the error in the cross-jumping jacks left.

For the cross-jumping jacks left the x-component mean error also has the biggest amount, with an absolute value of 19,6 cm of the error magnitude of 25 cm. The y-component mean error is 12,5 cm long and the z-component 5 cm.

Lastly, the regular jumping jacks have a mean magnitude error of 18,05 cm. The x-component error being 8 cm per frame long, y-component 14,02 cm and z-component mean error being 4,4 cm per frame long. So here the y-component shows the highest amount of error.

Taking all the mean value and highest error component amount in consideration one can deduce a correlation between prevalent movement on an axis and the prevalent axis error, as jumping jacks have the highest error on y, jumping jacks forward on y, and the cross jumping jacks being error prone to movement on the traversal plane.

## Case: Jumping Jacks Data-set, Constrained, Heel Joint

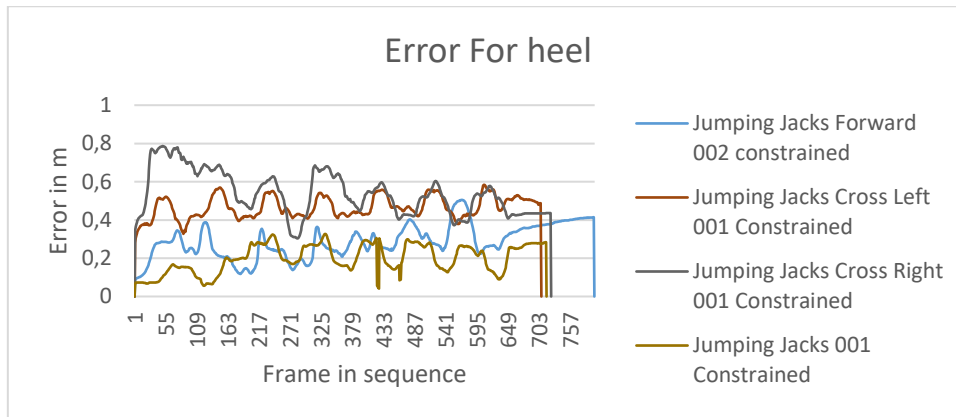


Figure 25: Error behavior for the left heel of the jumping jack data-set constrained

Lastly, the heel error of the jumping jacks data-set will be evaluated.

The number of the frame will be just as long as for the evaluation of the stays the same as in the analysis of the knee error for the jumping jacks, the frames do not change for the different joints.

The x-error on the heel ranges from 0 to 60 cm, where both cross-jumping jacks data-sets have locale maxima at 50 cm. They oscillate in with a wavelength of 100 frames and an amplitude of 30-40 cm for cross left and 20-40 cm for cross right.

The jumping jacks forward data-set showing low levels of oscillation on this axis, having a positive trend, though. It ranges from 10 cm error to 30 cm at the end of the motion.

Lastly, the regular jumping jacks show hints of oscillation, with a wavelength of approx. 100 frames and amplitude of sometimes 5 cm, other times 17 cm. For this data-set also a positive error trend is noticeable.

Next, the different data-sets of the y-error of the heel range from 0 to 70 cm. Where Jumping jacks cross right reaches the highest amount of the 4 error at approx. 70 cm at frame 60. It is trending negative, otherwise describing the plotted graph is hard to describe, as it is neither oscillating and not comparable to another obvious model.

On the other hand, the cross-jumping jacks left error is oscillation from 21 cm to 40 cm with a wavelength of around 90 frames.

The jumping jacks forward show signs of oscillation, however too vague to point to a specific value. The common jumping jacks also show signs of oscillation after frame 217, also as vague as for the forward jumping jacks. Before frame 217 this data-set the y-error is close to 0 cm for the jumping jacks constrained.

Next, the error on the z-axis shows highly frequent for all the four datasets. They range from 0 to 18 cm in error. The frequency of all the four data-set is 1 per 20 frames. The amplitudes are different for the data-set sets, however. For the jumping jacks forward, for example, has the

highest peak is at 18 cm at frame 97. Else the amplitude can also be 2 cm short. The other data-sets have similar behaving amplitudes ranging in these borders.

Finally, the magnitude of these four data-sets displays the error for the heel.

For the jumping jacks cross right, a slight negative trend and similarities to y-error for heel are noticeable. an oscillation is hard to find. The jumping jacks cross left graph shows periodic behavior every 100 frames. It ranges from 35 cm to 55 cm and does not trend in one direction.

Jumping jacks forward and the common jumping jacks don not show a sign of a trend, a multiple oscillation behavior can be deduced, on a very close look, every 100 frames, and every 20 frames.

The mean value of the jumping jack forward error is 28,70 cm, with an absolute x-component error of 19,7 cm, the absolute y-component error of 17,9 cm and an absolute z-component error of 4,5 cm. So, the x-error axis has the biggest amount of the error magnitude, y-component error not to ignore though.

The mean error magnitude of the jumping jacks is 46,4 cm. This data-set has the highest absolute component error on the y-axis with 32,86 cm. The x-component error coming second, with a mean of 28,44 cm and the z-component error mean being 6,7 cm.

The absolute mean error components of the x-component of the jumping jacks cross right data-set is 37,5 cm, for the y-component the mean error is 35,7 cm and for the absolute z-component 7,2. So the x-component has the highest mean error of the mean error magnitude of 53,8 cm.

Lastly, the mean magnitude error of the common jumping jacks is 19,8 cm, with a mean x-error component of 11,0 cm, mean y-error component of 12,8 cm and z-error component of 4,5 cm. So, the y-component has the highest mean amount of the magnitude error.

Here again, the correlation between movement on axis and error on the axis is visible, like at the knee movement.

### How is this Result to be Interpreted?

The error of measurement in meters is a value, that can be compared with other research, that has been made in this field. Svantesson and Bornold have compared motion tracking data with data that has been generated with the use of various inverse kinematics algorithms. They compared the distance of the measurement with the data that has been the output of the algorithm with each other and the computational time that an algorithm needs, to calculate the position of a marker. [7]

We can compare these values since joints have been set with a similar idea in mind. That wherever an ankle is, there will be one joint, and one bone of the human skeleton will map on the segment in the kinematic chain.

The comparison of the quality of the distance measurement data can take place with the Fabrik algorithm. For their constrained Fabrik algorithm they achieved the following results for a true position of an elbow in motion. Min value of 1 cm, a Max value of 56 cm, mean value of 45

cm and additionally Median, standard deviance and variance have been evaluated with a median of 46 cm, Std. Dev of 6 cm and Variance of 0 cm.

With the measurements, that have been taken in this thesis for a common jumping jack with constraints, see Figure 24, the statistics are similar. A min of 0 cm, a max value of 30 cm and a mean value of 18 cm for the left knee joint. Standard deviation is 5 cm and variance are 0.00 m.

For the left heel joints, the statistics are mean 19 cm. max 32 cm, min 0 cm, std. deviance is 7 cm and variance is 1 cm.

So, one can say, that the logarithmic scale is in the right size, however, the produced not inside of the measurements of the other standard deviance result. One also has to take in consideration, that the kinematic chain in Svantesson and Bornold work is longer and have been applied to the whole body. Deadlocks have also been a problem in the jumping jack measurement used for the statistical comparison.

Another work, with which this data can be compared is the study on arms by Parger et al. on Human upper-body inverse kinematics.

In their paper, they have modeled a kinematic chain with constraints to render arms in a virtual reality application. They have calculated the root means squared error (RSME), which is the chain of mathematical operation to calculate the magnitude of a vector and then calculate the mean of all the magnitudes. They modeled an analytical solution for upper body inverse kinematics, but also compared it to other available sources, the commercially available Final IK by RootMotion, the open source solution SAFullBodyIK by Stereo Arts, and fabrik with an unconstrained kinematic chain.

The unconstrained mean for jumping jacks knee 13 cm, 23 cm max, 0.00 cm min, 4 cm standard deviation, 0 cm variation. For heel, it is mean 11 cm, for max 20 cm, for min 0 cm, for the standard deviation 7 cm, for variation. 1 cm. Parger et al. have used different markers however their kinematic chain has the same length. For Fabrik their resulting data magnitude lies at 15.7 cm for URM, 27.4 cm for BM, 34,3 cm for MM and 29,6 cm. The acronyms mentioned refer to markers of an arm, a kinematic chain has still been used. [8]



## Discussion

### The Model

A four joint kinematic chain for the model is a fast-implemented solution with the default marker-sets offered by the Motive software.

As the marker-sets are optimized to their hardware and their use cases, a solution that requires less markers, however more accurate positions of the joints are not designated for this. A customized marker-set with marker-position at places optimized for an application of inverse kinematics algorithm might improve the accuracy of the positions with the Fabrik-algorithm and with other inverse kinematics algorithms.

The origin marker is close to static, the waist as an origin moves as well with a leg motion, so its inaccuracies are present.

A common origin for both kinematic chains could affect the model in an interesting way.

### The Algorithm

Fabrik for kinematic chains without constraints is a fast and easy implemented algorithm. It is well documented in the paper and other sources on the internet, the implementation for it did not take long. It can have legitimate use cases in for posing of skeletons. [18]

If this posing is enough for an integration into a motion tracking pipeline it can be researched further, since the use of more joints can help get a better result. However, with the use of the current model for a construction of a realistic leg motion using a target-tip-toe and origin-waist based it is not accurate enough not for satisfactory results.

Based on the kinematic chains with constraints used in this thesis the application of the algorithm on such a chain is problematic, due to deadlocks and the resulting break in distance constraints of the segments.

In general, the algorithm is computationally slim, compared to other algorithms used in research. [7]

Implementing more algorithms can help get a more rounded picture of the current state of the art of inverse kinematics algorithms.

Modifying the constraints in different directions gives better results, sticking to one combination of constraints will cause it to be good at one thing, but bad at something else

## Infrastructure

A first pipeline is functional, experience has been made implementing a kinematic model also on a conceptional level as well.

The BTK toolkit is an outdated toolkit, an update of the .c3d reader and writer would improve the system.

The .c3d format served well and an application in C++ is performant. There was some work done on the interface, but there is still some work to be done. A graphical interface, for inputting the data and choose the joints here, specify constraints and in the end press a button to calculate according to the algorithm one by one would be friendlier.

With the Vagrant virtual-machine a solution-oriented environment has been found, that can be setup on every operating system is compatible with vagrant.

Concerning the programming code, if a marker has been chosen, the values auto order, which is problematic, when using a marker, A as a joint, after another joint B, that comes in the marker-set comes earlier. The program would still create the kinematic chain as AB and not BA, as specified.

Also refactoring of the code should be performed. A solver and acquisition class have been included into the architecture of the application but the 1000 lines of code long main.cpp may not be up to current architecture standards. A good IDE is recommendable.

## Conclusion and Further work

All in all, a first acquisition of knowledge about the topic of motion tracking, inverse kinematics and their problems has been made. Individual position data can be compared in a visualization tool. The error of calculated position and its components can be output in a .csv data format and then displayed in forms of graphs.

The resulting motion is reconstructed quickly; however, the accuracy is quite disappointing and that is an important aspect. The fact that the algorithms application on kinematic chains with joints increases computational time and worsens the accuracy is also problematic.

For further research the CCD-Algorithm would be an interesting addition to environment to be able to compare the results of algorithms with each other. This would be a good basis to find and research methods to improve the result.

## Appendix A: Parameters for the Inverse Kinematics Solving Program

The written Program allows for executing it in different variations and pre-configuration, a table of the programmed parameters are displayed in Table 3.

|   |  |
|---|--|
| -u  | The solving algorithm will be executed without constraints   |
| -c  | The solving algorithm will be executed with constraints  |
| -s  | The solving algorithm will generate an origin between both first chosen joints                             |
| -t  | The solving algorithm will be executed in the default setting, with test input data and preselected joints |
| --joints -left={<joint_id>;...;< joint_id>}   | Joint specification for the left leg are defined   |
| --joints -right={< joint_id>;...;< joint_id>}   | Joint specification for the right leg are defined  |
| --joints-constraints-right={< joint_id>-> [<angle_constraint_up>, <angle_constraint_down>, <angle_constraint_left>, <angle_constraint_right>; ... ; < joint_id >-> [<angle_constraint_up>, <angle_constraint_down>, <angle_constraint_left>, <angle_constraint_right>]} | Joint and constraint specifications will be predefined   |
| --joints-constraints-left={< joint_id>-> [<angle_constraint_up>, <angle_constraint_down>, <angle_constraint_left>, <angle_constraint_right>; ... ; < joint_id >-> [<angle_constraint_up>, <angle_constraint_down>, <angle_constraint_left>, <angle_constraint_right>]}  | Joint and constraint specifications will be predefined   |

Table 3: Parameter Specifications for the Command line interface (CLI)

The placeholder <joint\_id> needs to be replaced with the position in order of the joint and <angle\_constraint\_up>, <angle\_constraint\_down>, <angle\_constraint\_left>, <angle\_constraint\_right> with the corresponding angle values in degrees.

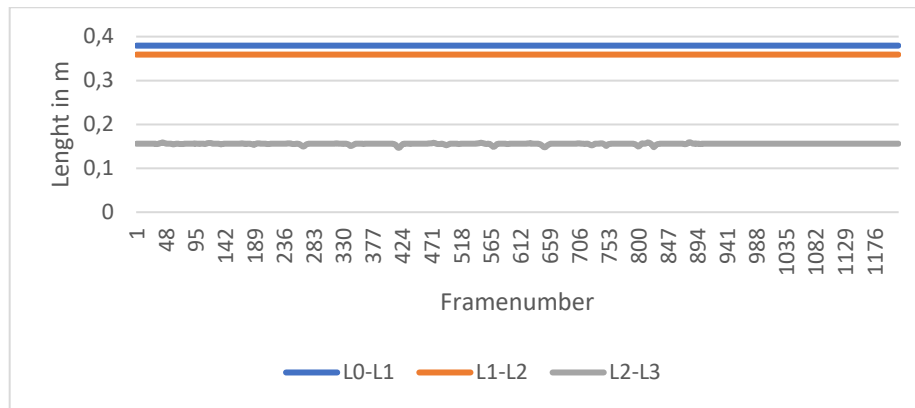
Another important note about the prespecification of joints and joints with constraints is that the use of symbols is important for the reading process, using a ‘,’ to separate joint id from each other will result in errors in the solving process, just as the use of ‘;’ to separate angle constraints. A comma (‘,’) needs to be used to separate the angle constraints and a semicolon to separate the joint id s from each other. The symbols in the parameter list are important for the correct use of the parameters, that also accounts for the pointer between id and constraint ‘->’ and the kinds of parenthesis.

## Appendix B: Rizzoli Marker Descriptions [15]

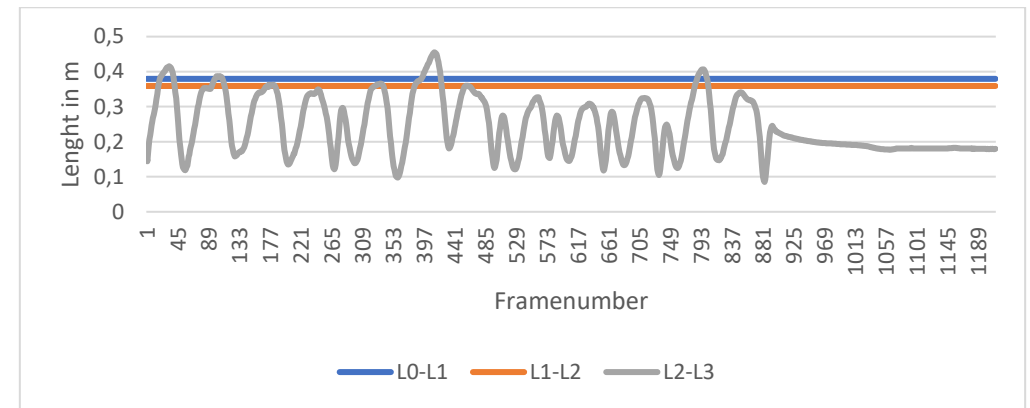
| Rizzoli Lower Body Protocol Markers |                 |  |
|-------------------------------------|-----------------|--|
| Label                               | Related Segment | Description  |
| RASIS                               | Pelvis          | Anterior superior iliac spine.   |
| LASIS                               |                 |  |
| RPSIS                               | Pelvis          | Posterior superior iliac spine.  |
| LPSIS                               |                 |  |
| RGT                                 | Upper Leg       | Most lateral prominence of the greater trochanter external surface.  |
| LGT                                 |                 |  |
| RTH                                 | Upper Leg       | Place near the midline of the thigh. Used only for a tracking purpose of distinguishing left and right side. For best result, offset the height of the marker between left and right side.   |
| LTH                                 |                 |  |
| RLE                                 | Upper Leg       | Most lateral prominence of the lateral femoral epicondyle. Together with LM markers, it determines the location of knee joint axis.  |
| LLE                                 |                 |  |
| RHF                                 | Lower Leg       | Head, proximal end, of the fibula.   |
| LHF                                 |                 |  |
| RTT                                 | Lower Leg       | Most anterior border of the tibial tuberosity.   |
| LTT                                 |                 |  |
| RSK                                 | Upper Leg       | Place near the midline of the shin. Used only for a tracking purpose of distinguishing left and right side. For best result, offset the height of the marker between left and right side.  |
| LSK                                 |                 |  |
| RLM                                 | Lower Leg       | Distal apex of the lateral malleolus.  |
| LLM                                 |                 |  |
| RCA                                 | Foot            | Upper ridge of the calcaneus posterior surface. The aspect of the Achilles tendon insertion on the calcaneus <sup>[1]</sup> .  |
| LCA                                 |                 |  |
| RVM                                 | Foot            | Dorsal aspect of fifth metatarsal head.  |
| LVM                                 |                 |  |
| RFM                                 | Foot            | Dorsal aspect of first metatarsal head.  |
| LFM                                 |                 |  |
| RDP1                                | Foot            | These markers are added on the distal phalanx only for the toe segment tracking purpose, and they are not included in the biomechanical analysis. Place the marker near the end of the big toe. More specifically, the marker should be placed at the distal end of the first phalanges. |
| LDP1                                |                 |  |
| RME*                                | Upper Leg       | Medial prominence of the medial femoral epicondyle.  |
| LME*                                |                 |  |
| RMM*                                | Lower Leg       | Distal apex of the medial malleolus.   |
| LMM*                                |                 |  |
| RSM*                                | Foot            | Dorsal aspect of second metatarsal head.   |
| LSM                                 |                 |  |

Table 4: Acronyms for Markers for the Rizzoli protocol

## Appendix C: Measurement Tables



(a)



(b)

Figure 17: Distances between joints for (a) unconstrained; (b) constrained measurements

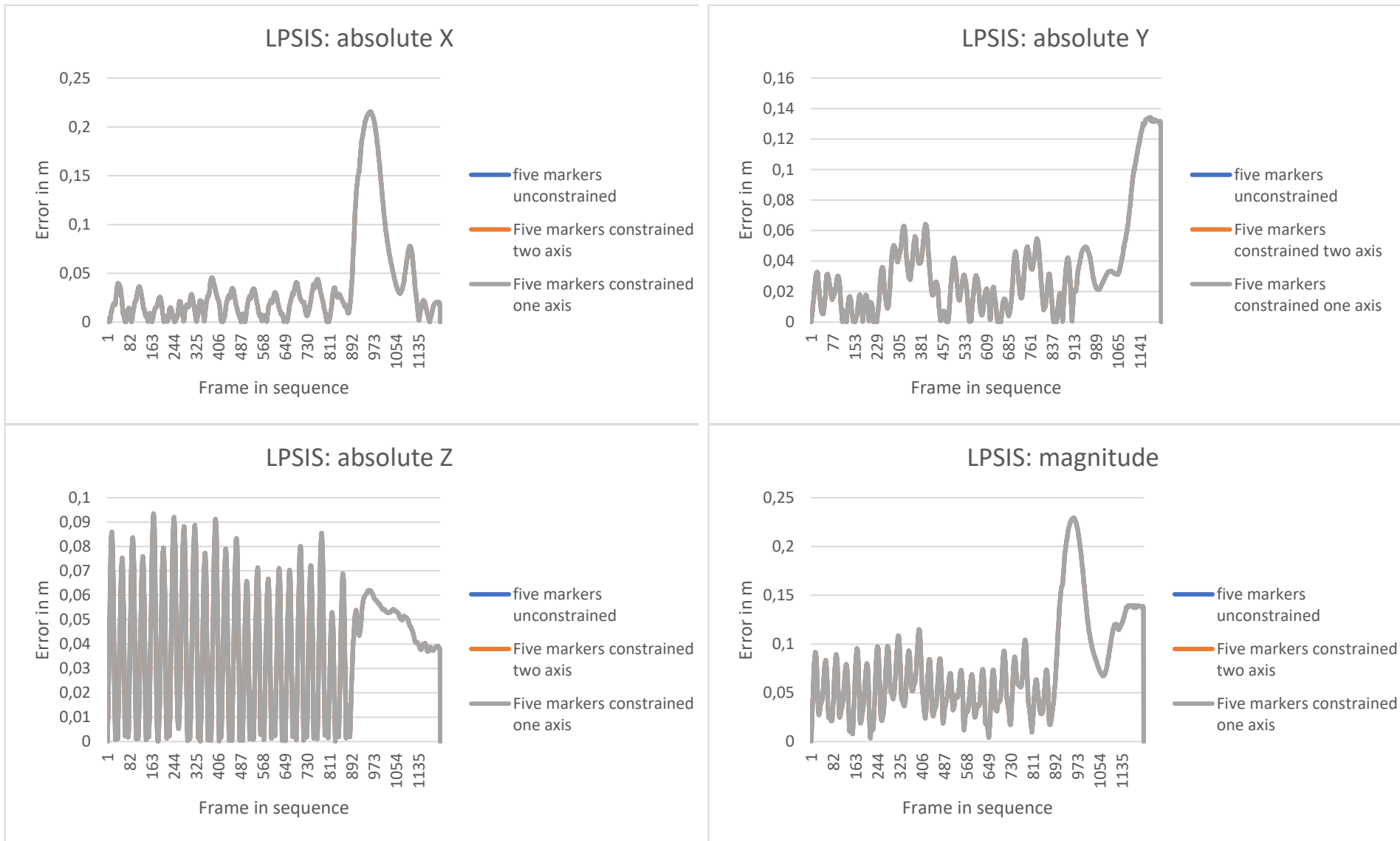


Figure 18: Error behavior for LPSIS marker for the jog data-set

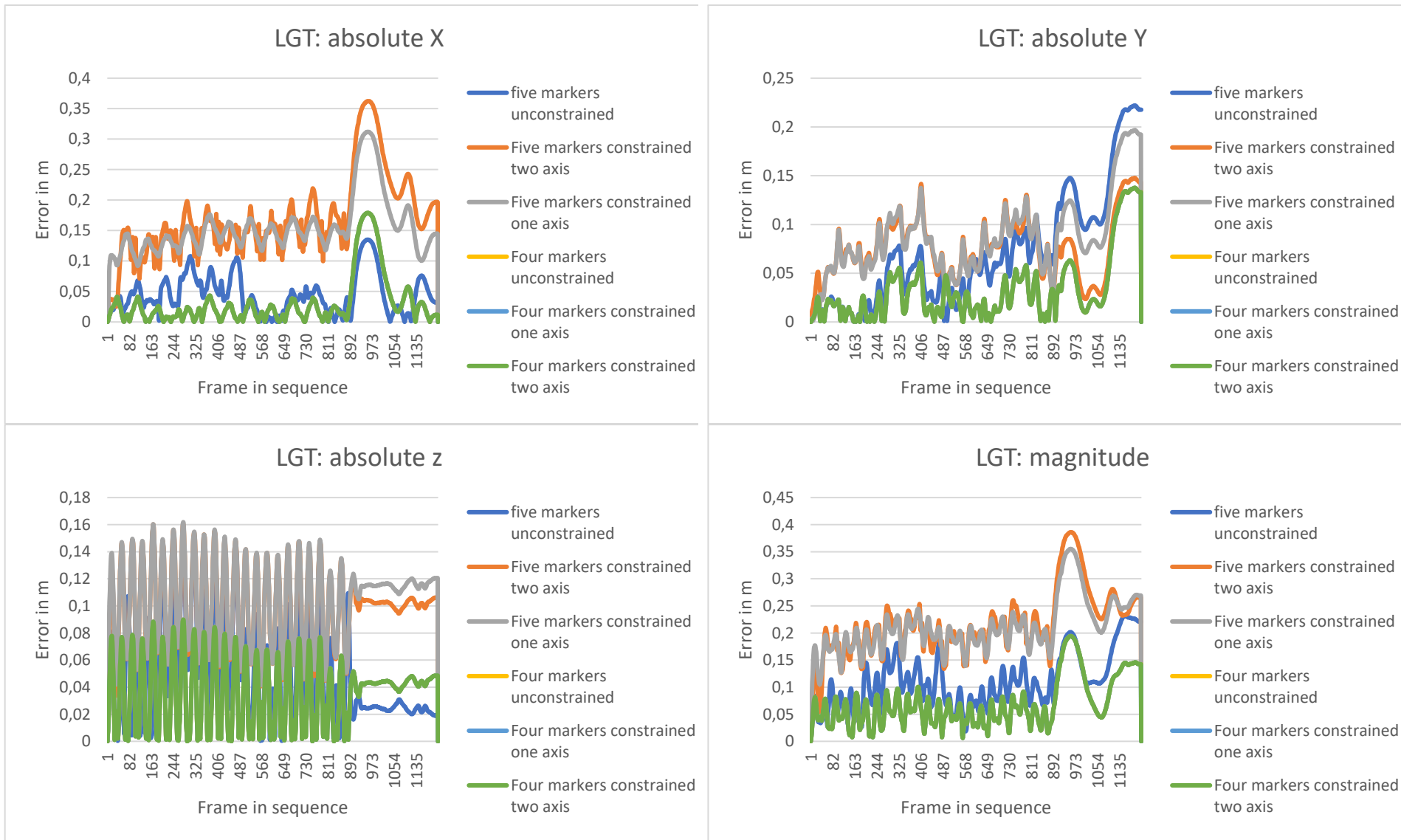


Figure 19: Error behavior for LGT marker for the jog data-set



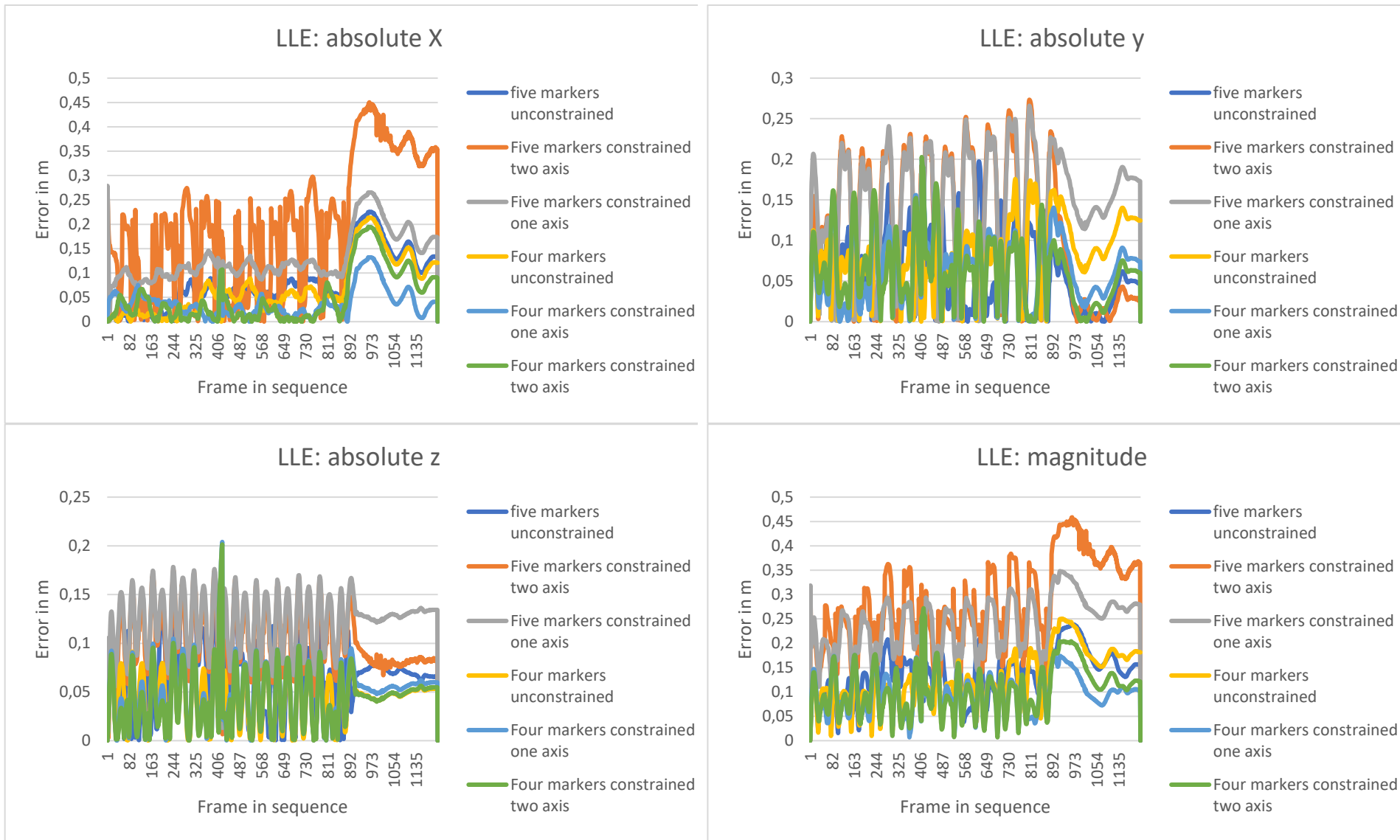


Figure 20: Error behavior for LLE marker for the jog data-set

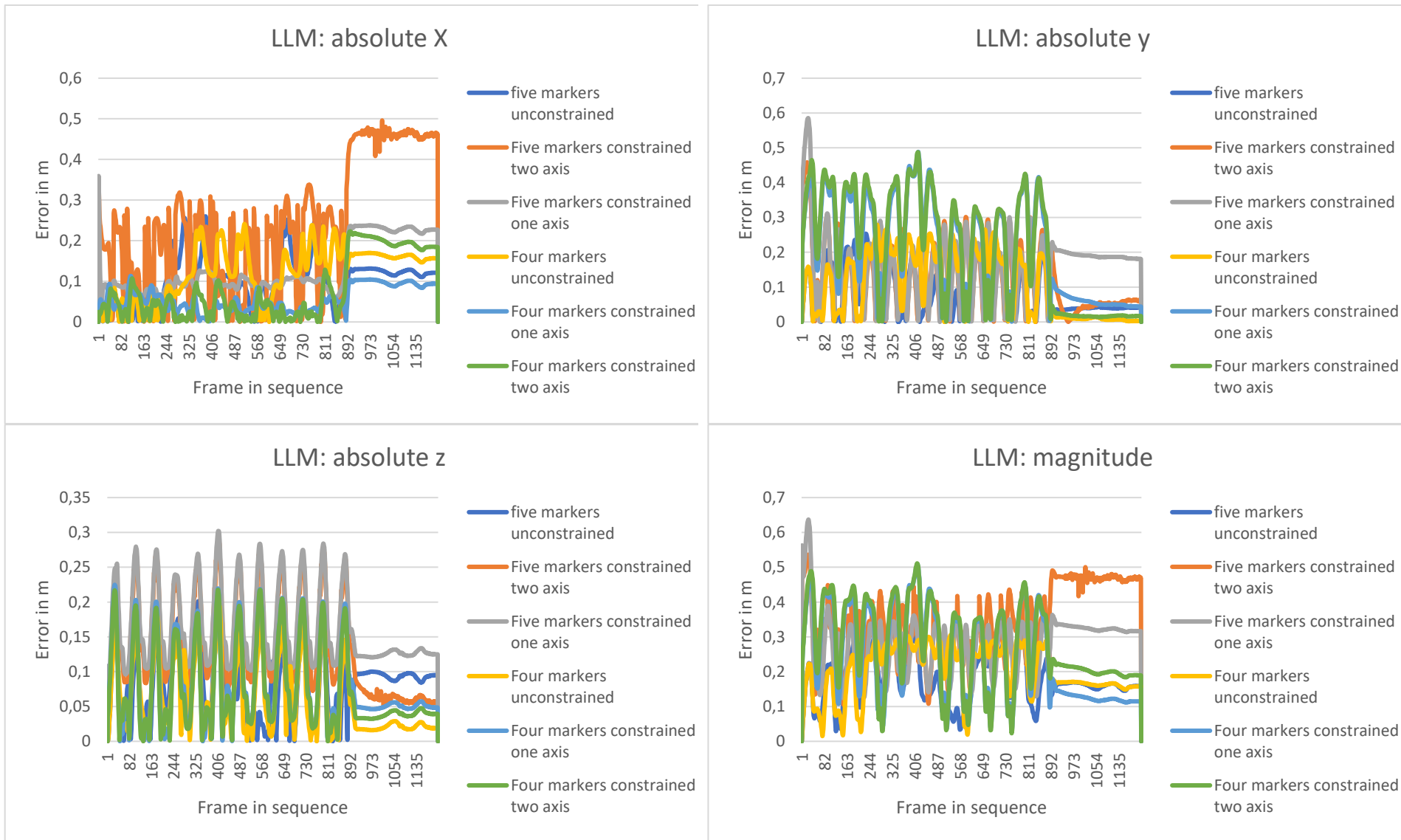


Figure 21: Error behavior for LLM marker for the jog data-set

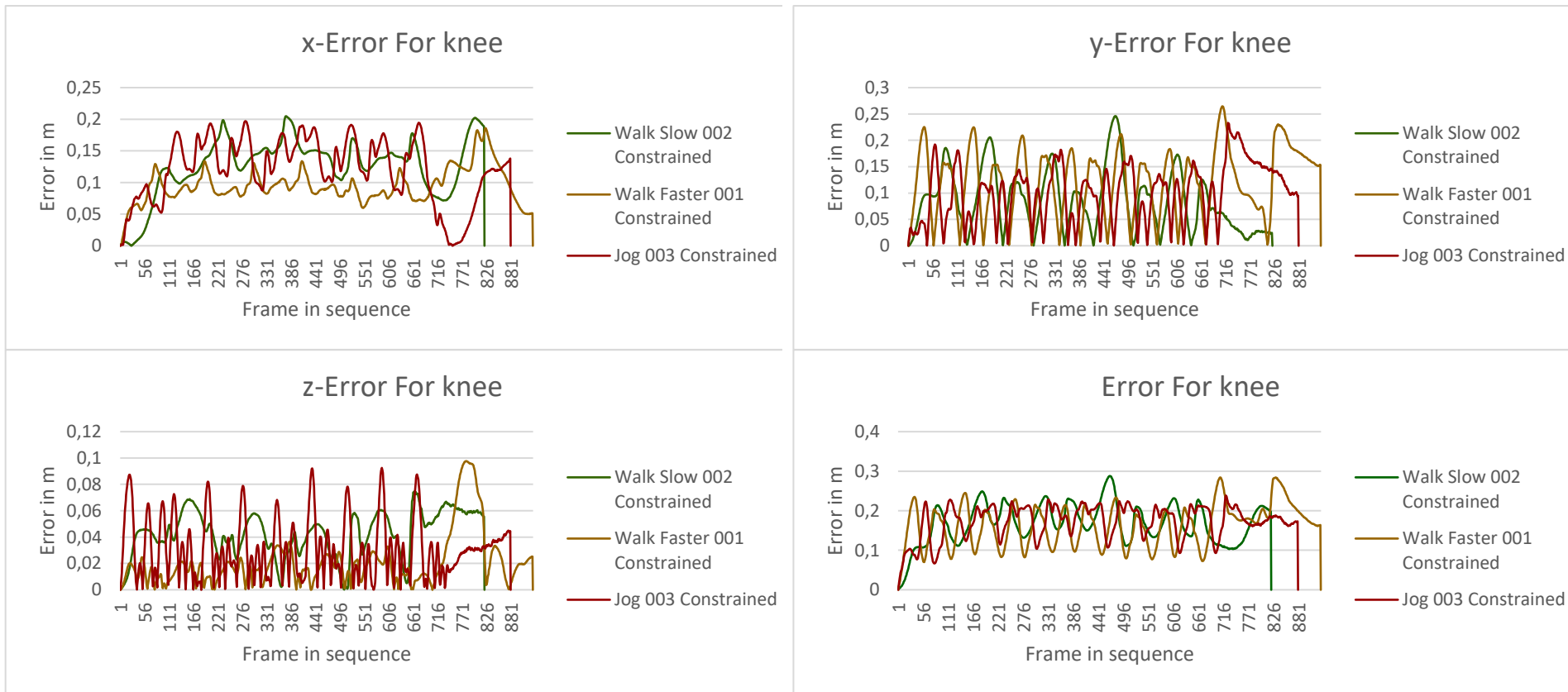


Figure 22: Error behavior for the left knee of the treadmill data-set constrained

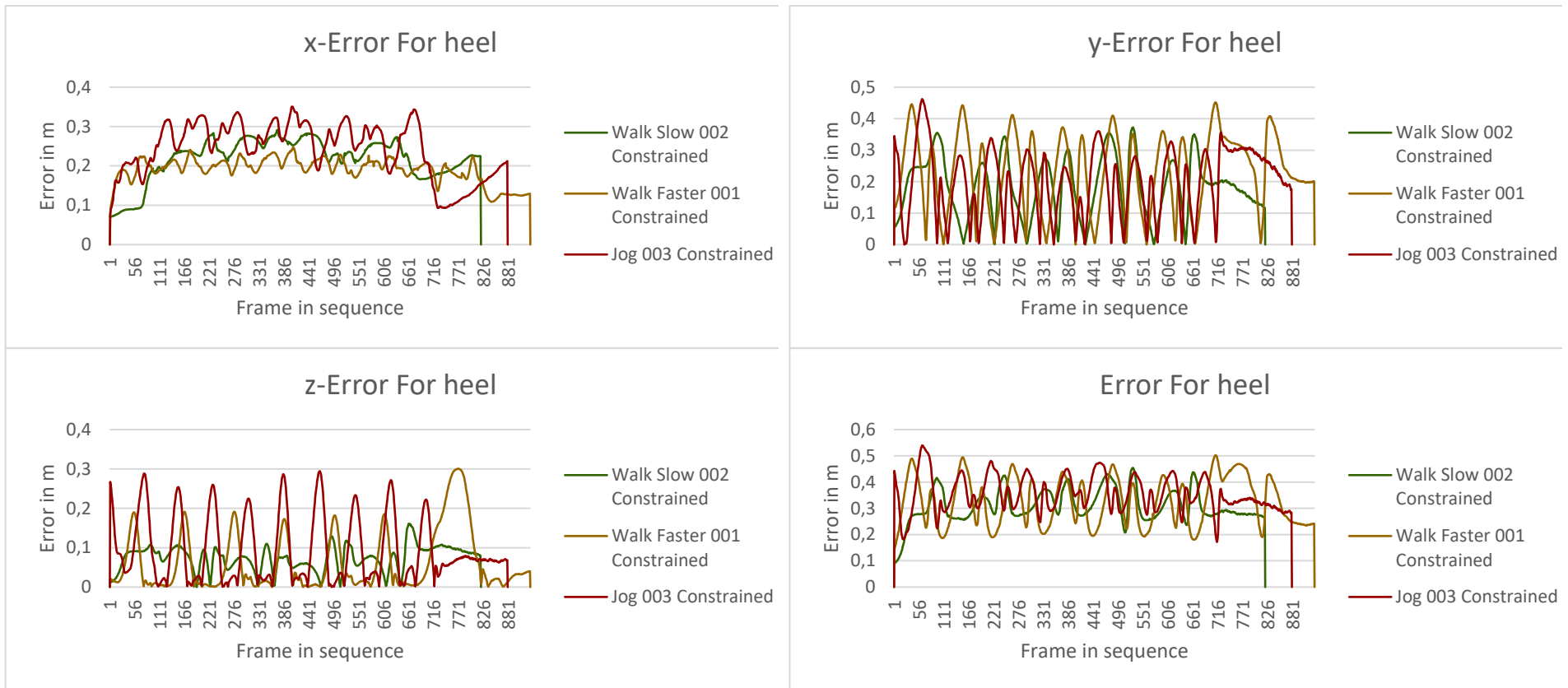


Figure 23: Error behavior for the left heel of the treadmill data-set constrained

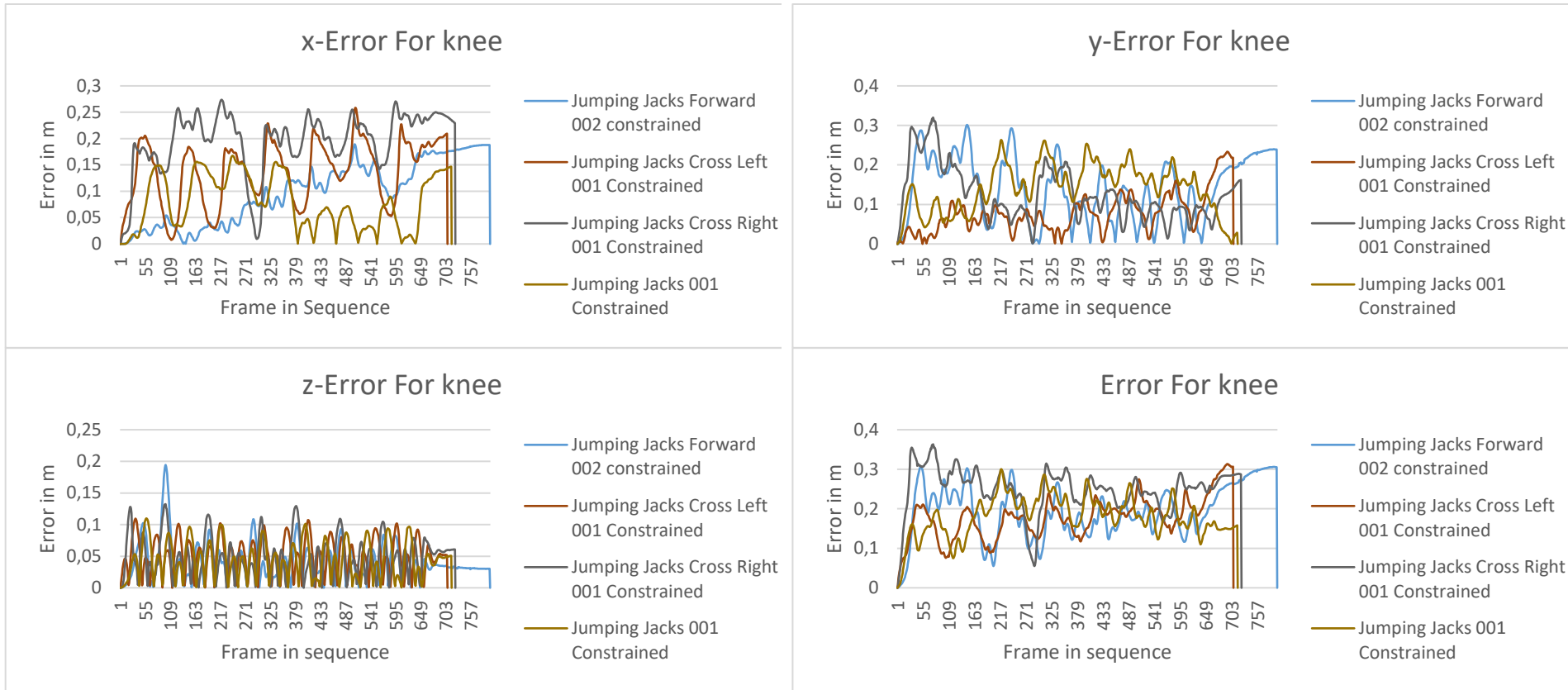


Figure 24: Error behavior for the left knee of the jumping jack data-set constrained

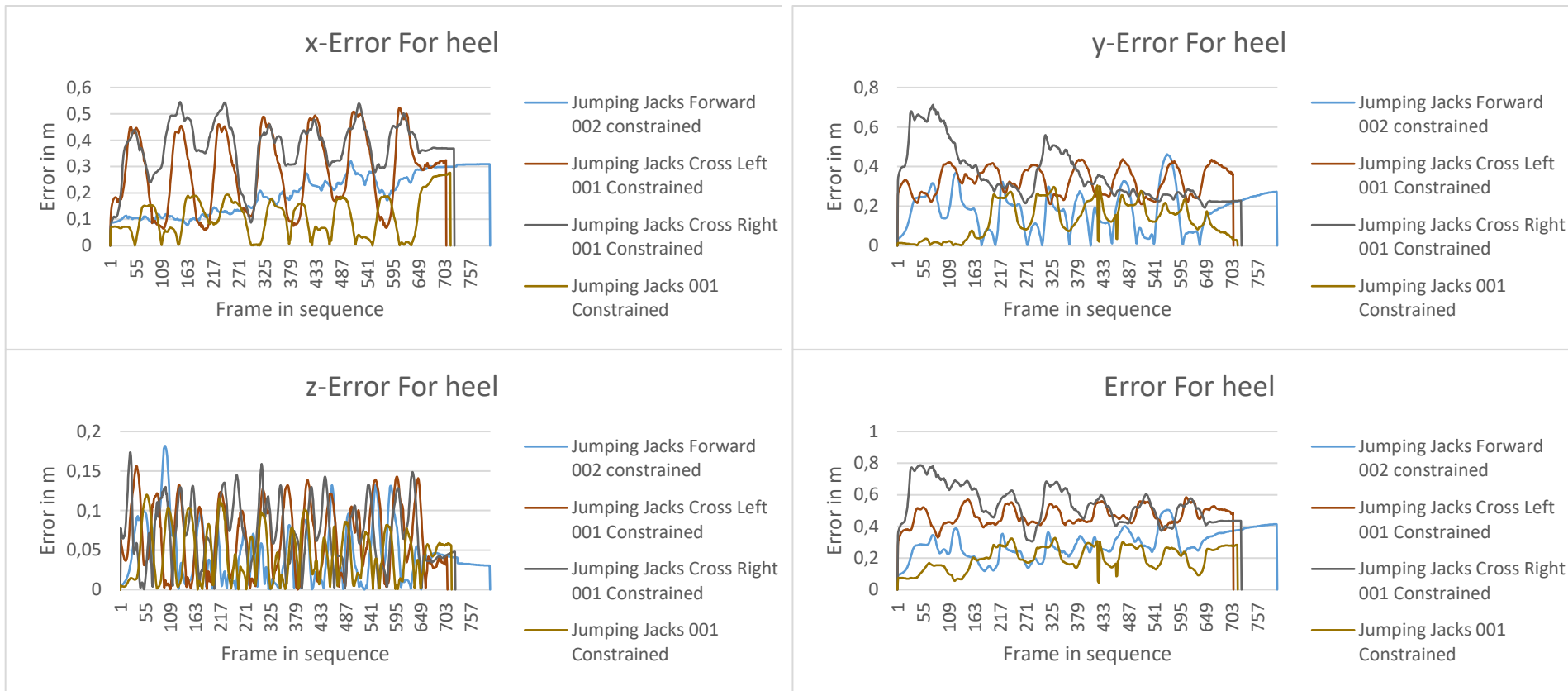


Figure 25: Error behavior for the left heel of the jumping jack data-set constrained

## Table of Source

- [1] M. Girard und A. A. Maciejewski, „Computational Modeling for the Computer Animation of Legged Figures,“ in *ACM SIGGRAPH Computer Graphics*, San Francisco, 1985.
- [2] S. Głowiński und T. Krzyżyński, „AN INVERSE KINEMATIC ALGORITHM FOR THE HUMAN LEG,“ *Journal of theoretical and applied mechanics*, pp. 53-61, 1 January 2016.
- [3] J. M. Wang, S. R. Hamner, S. L. Delp und V. Koltun, „Optimizing Locomotion Controllers Using Biologically-Based Actuators and Object,“ *ACM Transactions on Graphics (TOG)*, 31 4 2012.
- [4] J. Markowitz und H. Herr, „Human Leg Model Predicts Muscle Forces, States, and Energetics during Walking,“ *Computational Biology*, pp. 1-30, 13 Mai 2016.
- [5] L.-C. T. Wang und C. C. Chen, „A Combined Optimization for Solving the Inverse Kinematics Problem of Mechanical Manipulators,“ *Transactions on robotics and automation*, pp. 489-499, 1 August 1991.
- [6] A. Aristidou und J. Lasenby, „FABRIK: A fast, iterative solver for the Inverse Kinematics problem,“ *Graphical Models*, pp. 243-260, 15 May 2011.
- [7] J. Svantesson und J. Bornold, „A Real-Time Adaptation of Inverse Kinematics for Motion Capture,“ 1 June 2015. [Online]. Available: <http://publications.lib.chalmers.se/records/fulltext/219734/219734.pdf>. [Zugriff am 19 April 2019].
- [8] M. Parger, D. Schmalstieg, J. H. Mueller und M. Steinberger, „Human Upper-Body Inverse Kinematics for Increase Embodiment in Consumer-Grade Virtual Reality,“ in *VRST*, Tokyo, Japan, 2018.
- [9] P. K. Artemiadis, P. T. Katsiaris und K. J. Kyriakopoulos, „A biomimetic approach to inverse kinematics for a redundant robot arm,“ *Autonomous Robots*, pp. 293-308, 1 November 2010.
- [10] C. E. Hrabia, K. Wolf und M. Wilhelm, „Whole hand modeling using 8 wearable sensors: biomechanics for hand pose prediction,“ in *AH '13 Proceedings of the 4th Augmented Human International Conference*, Stuttgart, Germany, 2013.
- [11] Edoarado, „Wikimedia Commons,“ 13 November 2012. [Online]. Available: [https://commons.wikimedia.org/wiki/User:Edoarado#/media/File:Anatomical\\_Planes-en.svg](https://commons.wikimedia.org/wiki/User:Edoarado#/media/File:Anatomical_Planes-en.svg). [Zugriff am 12 April 2019].
- [12] „OptiTrack,“ NaturalPoint Inc., 2019. [Online]. Available: <https://optitrack.com>. [Zugriff am 12 April 2019].

- [13] E. Cramp, „c3d.org,“ Motion Lab Systems, 23 January 2013. [Online]. Available: <https://www.c3d.org/index.html>. [Zugriff am 12 April 2019].
- [14] A. ". Barré, „Mokka - Motion Kinematic ; Kinetic Analyzer,“ The Biomechanical ToolKit (BTK), 2013. [Online]. Available: <http://biomechanical-toolkit.github.io/mokka/>. [Zugriff am 12 April 2019].
- [15] „OptiTrack Documentation Wiki,“ NaturalPoint Inc., 9 November 2018. [Online]. Available: [https://v21.wiki.optitrack.com/index.php?title=OptiTrack\\_Documentation\\_Wiki](https://v21.wiki.optitrack.com/index.php?title=OptiTrack_Documentation_Wiki). [Zugriff am 11 April 2019].
- [16] e. a. Leardini A, „A new anatomically based protocol for gait analysis in children,“ *Gait & Posture*, pp. 560-571, 1 October 2007.
- [17] A. ". Barré, „BTK Documentation,“ The Biomechanical ToolKit (BTK), 27 Dezember 2013. [Online]. Available: <http://biomechanical-toolkit.github.io/docs/API/index.html>. [Zugriff am 11 April 2019].
- [18] EgoMoose, „FABRIK (Inverse kinematics),“ 2 June 2016. [Online]. Available: <https://www.youtube.com/watch?v=UNoX65PRehA>. [Zugriff am 12 April 2019].
- [19] M. Y. Hassan und S. S. Ghintab, „Modeling and Position Control of Human Lower Limb Rehabilitation Robot using Pneumatic Muscle Actuators,“ *Al-Mansour Journal*, pp. 61-86, 2015.
- [20] W. C. Hsu, T. Sugiarto, Y. J. Lin, F. C. Yang, Z. Y. Lin, C. T. Sun, C. L. Hsu und K. N. Chou, „Multiple-Wearable-Sensor-Based Gait Classification and Analysis in Patients with Neurological Disorders,“ *sensors*, pp. 1-17, 11 Oktober 2018.
- [21] HashiCorp, „vagrantup.com,“ HashiCorp, 9 April 2019. [Online]. Available: <https://www.vagrantup.com/docs/index.html>. [Zugriff am 11 April 2019].
- [22] L. Unzueta, M. Peinado, R. Boulic und Á. Suescun, „Full-body performance animation with Sequential Inverse Kinematics,“ *Graphical Models*, pp. 87 - 104, 8 April 2008.



## Table of Figures

|  |    |
|--|----|
| Figure 1: Planes of motion [11].....   | 4  |
| Figure 2: Lower body model with legs separated, unconstrained.....   | 6  |
| Figure 3: Lower body model, without constraints, with generated sacrum in between.....   | 7  |
| Figure 4: Single link, with constraint.....  | 7  |
| Figure 5: Lower body model for legs separated, with constraints.....   | 8  |
| Figure 6: Kinematic chain solved by Fabrik algorithm [6].....  | 9  |
| Figure 7: Baseline Lower Body marker-set.....  | 13 |
| Figure 8: Rizzoli Lower Body Protocol.....   | 14 |
| Figure 9: File size of the bvh format.....   | 15 |
| Figure 10: File size of the c3d format.....  | 15 |
| Figure 11: Flowchart of the program.....   | 16 |
| Figure 12: Visualisation of the result/unconstrained/jog_001.c3d dataset in Mokka, only the markers WaistFrontLeft, LKneeOut, LHeel and LToeTip are enabled.....   | 18 |
| Figure 13: Eighteen Cameras and three hubs, Motion tracking setup.....   | 19 |
| Figure 14: Trajectories of the joints for a jogging motion with a lower body baseline marker-set for (a),(b),(c) and for Rizzoli lower body protocol (d),(e),(f).....  | 21 |
| Figure 15: Segments lengths for (a) unconstrained kinematic chain and (b) constrained.....   | 22 |
| Figure 16: Delta value for jogging motion in different variant, with the Rizzoli protocol.....   | 23 |
| Figure 17: Distances between joints for (a) unconstrained; (b) constrained measurements....  | 37 |
| Figure 18: Error behavior for LPSIS marker for the jog data-set.....   | 38 |
| Figure 19: Error behavior for LGT marker for the jog data-set.....   | 39 |
| Figure 20: Error behavior for LLE marker for the jog data-set.....   | 40 |
| Figure 21: Error behavior for LLM marker for the jog data-set.....   | 41 |
| Figure 22: Error behavior for the left knee of the treadmill data-set constrained.....   | 42 |
| Figure 23: Error behavior for the left heel of the treadmill data-set constrained.....   | 43 |
| Figure 24: Error behavior for the left knee of the jumping jack data-set constrained.....  | 44 |
| Figure 25: Error behavior for the left heel of the jumping jack data-set constrained.....  | 45 |
| <br>   |    |
| Table 1: Categorization of knee joint movement on planes of motion.....  | 12 |
| Table 2: (a) Constraints for a kinematic chain of 5 elements, Rizzoli protocol, left leg, with two axes used, (b) Constraints for a kinematic chain of 5 elements, Rizzoli protocol, left leg, with two axes used..... | 24 |
| Table 3: Parameter Specifications for the Command line interface (CLI).....  | 35 |
| Table 4: Acronyms for Markers for the Rizzoli protocol.....  | 36 |
| <br>   |    |
| Interface 1: Usage of the application written.....   | 16 |
| Interface 2: Log of an when opening, saving and reloading an c3d export Optitrack style.....   | 20 |

## **Eigenständigkeitserklärung**

Hiermit versichere ich, dass ich die vorliegende Bachelor-Thesis mit dem Titel:

**An Inverse Kinematics Model for the lower body Motion: Implementation of the Fabrik-Algorithm as part of a Motion Tracking Pipeline**

Selbständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich wörtlich aus der Literatur oder aus anderen Quellen wie z.B. Internetseiten übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

---