Appendix 2

Transformation & Scaling Scripts to Register Muscle Force Vector Data

Personalised HTO has mechanical safety equivalent to generic HTO: findings from a novel case-control in silico clinical trial. MacLeod et al

# System requirements & Installation

The software requires Matlab (The MathWorks, Inc., Natick, USA) to run. Matlab can run on various operating systems[[1]](#footnote-1) with a standard desktop computer. The software was recently tested on Matlab 2020a without any issues and we would thus recommend using the latest version of Matlab.

The software has only been tested using Windows 10, 64 bit, operating system and Matlab version 2020a, and we cannot guarantee compatibility.

No installation is required for the scripts themselves, they can be run immediately within Matlab environment.

# Instructions for Scripts and Demo data

There are three key steps with associated scripts to run which are prefixed with RUN\_ in the relevant folder. All the demo data required to run the scripts is provided. If you are familiar with Matlab, just run each script in order. If you are unfamiliar with Matlab, follow the instructions below.

## Part 1 – Registration

The script RUN\_Opt\_Trans\_rand.m is intended to register a set of points to a new orientation. It is currently set-up to use a matching set of 5 registration points in the form [;]. In this demo these points were obtained from two similar 3D models in different orientations. The original model points are OS\_pts\_5.mat and the new model points are P28\_pts.mat. The script will determine the transformation required to match the points from the original model to the new model.

1. Navigate to the folder: “Step 1 – Registration”
2. Load the data points to perform the registration by typing into the Matlab command line:

Load(‘OS\_pts\_5.mat’)

Load(‘P28\_pts.mat’)

1. Run the script

RUN\_Opt\_Trans\_rand\_5pts\_sc

1. When it runs it will request “*enter the points to be alligned to:*”

Enter: P28\_pts

1. It will then request: “enter the points to be alligned/transformed:

Enter: OS\_pts\_5

1. The optimisation routine will run. This typically takes under 10 seconds to complete.
2. Once complete, it outputs the calculated transformation matrix (trans) and displays the transformed points in a figure.

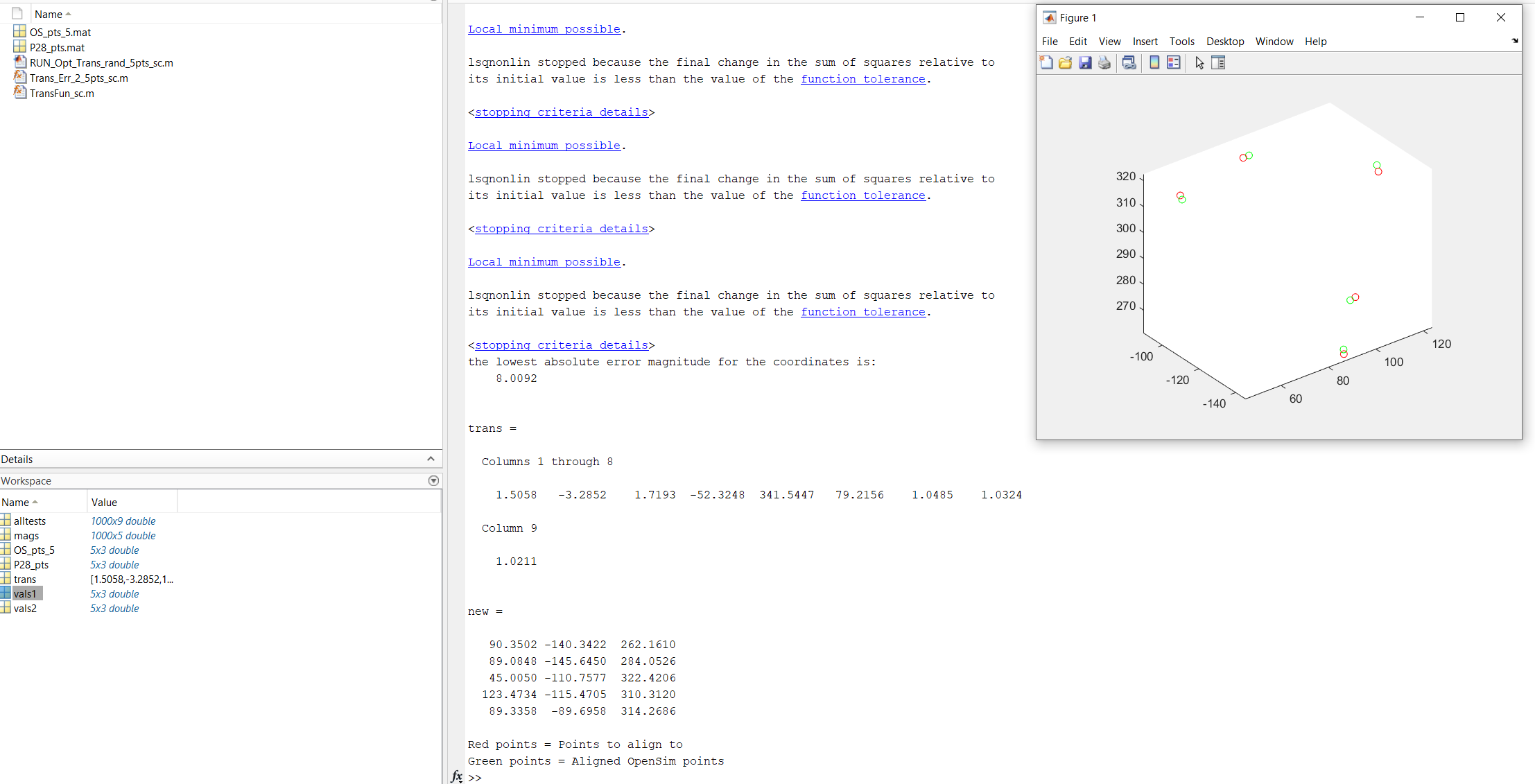


Figure - Output after running the “RUN\_Opt\_Trans\_rand\_5pts\_sc.m” Matlab script.

## Part 2 – Transform Muscle Forces

The second script, RUN\_multiply\_gait\_mags\_write\_csv\_ALLPATIENTS, uses the previously calculated transformation matrix, trans, to re-orientate a set of existing muscle force vector data to the coordinate system of the new model.

1. Navigate to the folder: “Step 2 – Transform Muscle Forces”
2. It requires the following variables to be provided as input files in .mat format:

Ivecs – the input vectors for the muscle forces at each required instance of the activity.

Imags - the input magnitudes for the muscle forces at each required instance of the activity.

Timepoints – a list of the time points at which the activity instance occurs for ordering and referencing the instances.

trans – the transformation matrix

Heights\_Weights.mat - this file was used as a look-up table for patient weight. For this demo all data has been redacted other than those required for the relevant demo case.

It also requires the function: TransFunRotOnly.m to be present within the folder.

1. The script demo is expected to run in under a few seconds on a "normal" desktop computer
2. The expected output is 5 .csv files labelled with the activity instance containing the transformed muscle force vectors (Figure 2).

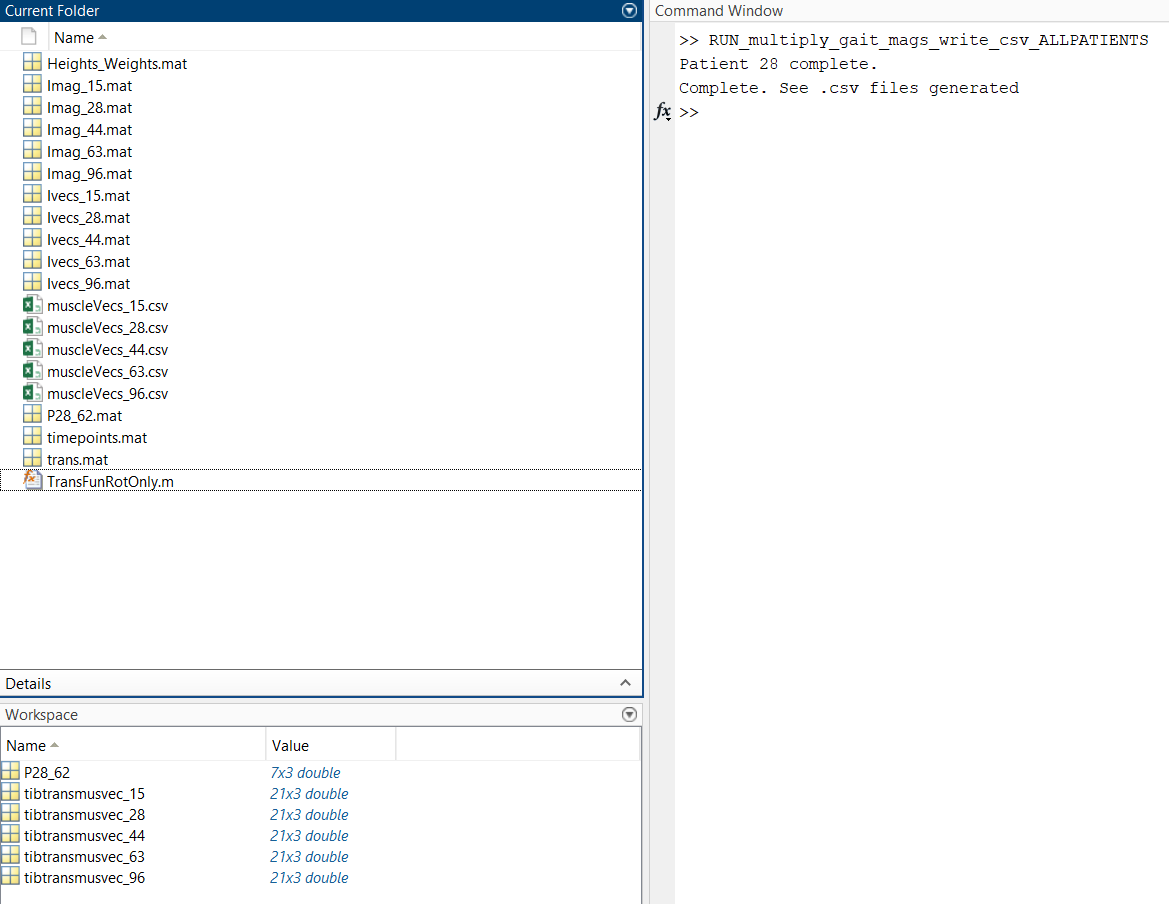


Figure – The output .csv files

## Step 3 – Transform Joint Reaction Forces

The script RUN\_Basic\_JointR\_cycle.m uses existing joint reaction force vector data, selects the values at key instances and transforms then. It also determines the location across the joint line where the force is applied using a proportion of the distance between specific landmarks on the left and right side of the knee.

1. Navigate to the folder: “Step 3 – Transform Joint Forces”
2. It requires the following variables to be provided as input files in .mat format:

Lat100 – the input vector for the lateral joint forces at each required instance of the activity.

Med100 – the input vector for the medial joint forces at each required instance of the activity.

timepoints – a list of the time points at which the activity instance occurs for ordering and referencing the instances.

trans – the transformation matrix

Heights\_Weights.mat - this file was used as a look-up table for patient weight. For this demo all data has been redacted other than those required for the relevant demo case.

P28\_62.mat – these are the landmark coordinates used to determine the location of the joint forces.

It also requires the function: TransFunRotOnly.m to be present within the folder.

1. The script outputs several .csv files: MedJointLoc.csv, LatJointLoc.csv, and Timepoints.csv.

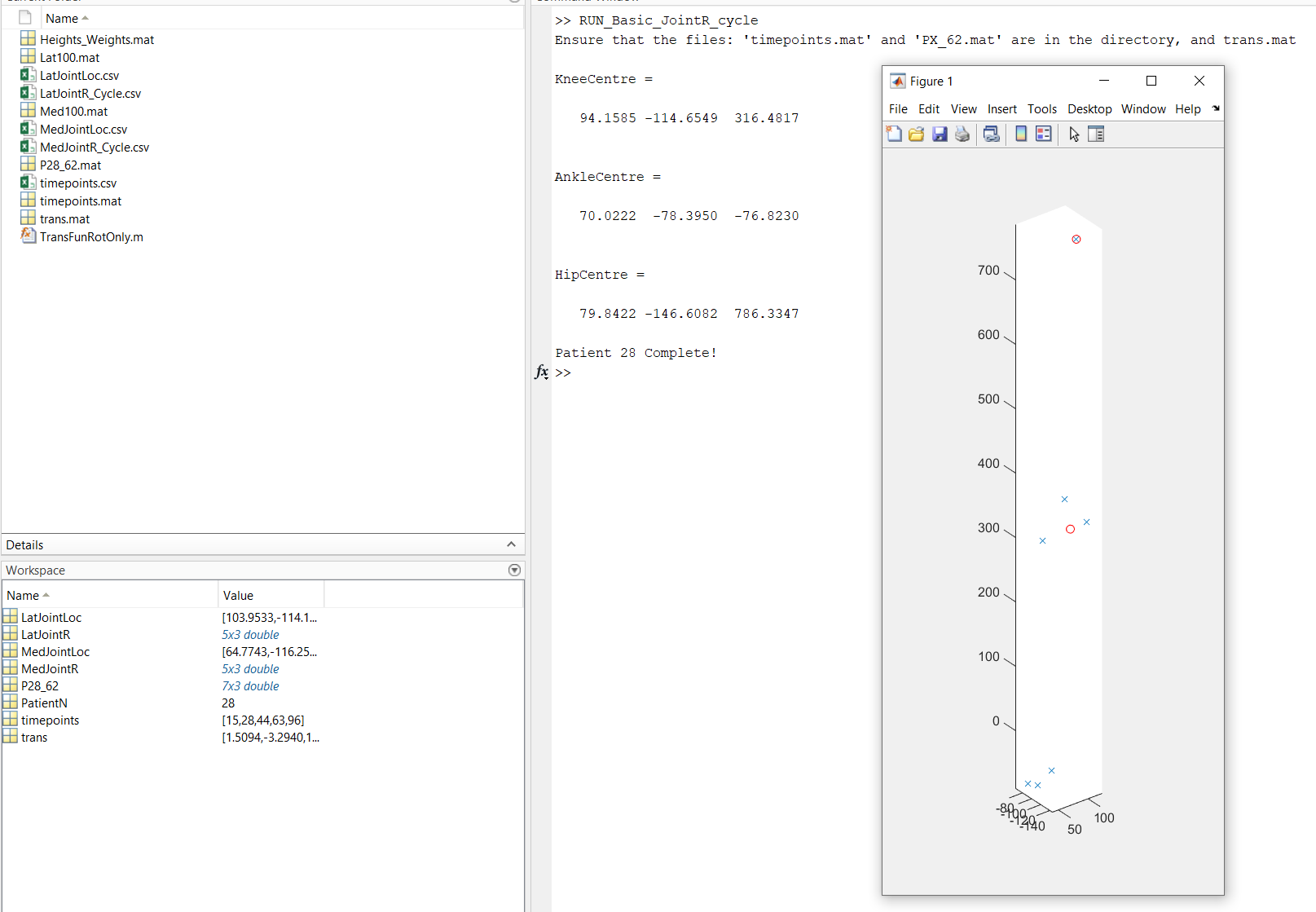


Figure - Displayed plot showing the landmarks used for the determination of the joint forces

1. https://uk.mathworks.com/support/requirements/matlab-system-requirements.html [↑](#footnote-ref-1)