Scilab project

Martin Le Guennec

2023/05/05

Table des matières

1. Description of the code

This code compares the Vicon position data with acceleration data associated with angular velocity data from K-Invent's K-Move IMU.

All the useful information to understand and run this code are contained in the README.md file.

All the scripts contained in this project were coded following the guidelines provided by Richard Johnson: https://www.ee.columbia.edu/~marios/matlab/MatlabStyle1p5.pdf

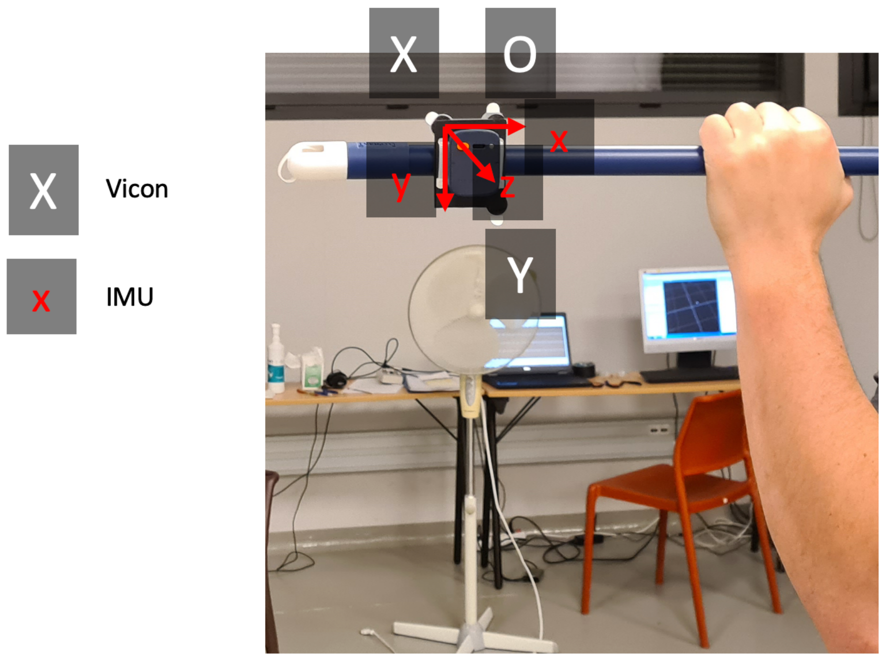
2. Context

The objective of this code is to validate the K-Invent IMU as a good estimator of velocity. Their aim is to use the K-Move (their IMU) in Strength and Conditioning as a tool to make VBT (Velocity Based Training). This training method uses the velocity at which the barbell is lifted rather than the weight; it is supposed to be a better training variable for assessing the neuromuscular fatigue and individualize training.

For this purpose, I measured 3 set of 8 squats at different speed: slow, average and fast.

2.1. Treated data

For each set, the squat was measured with the Vicon and with the K-Move as follows.



The two devices were synchronized thanks to the Delsys. The acquisition was began with the Vicon, we started the measurement with this device and it set on the Delsys that consisted of a constant signal. Then, when we started the measurement of the K-Move, it sent a signal to the Delsys. Therefore, we can synchronize the two devices together with this last signal.

3. Execution of the code

To run the code, you just have to open de main.sce in Scilab and run it. All the scripts will execute, and you will find the figures saved in the RES folder.

3.1. Scripts

main.sce

The main.sce script organizes the workspace by calling InitTRT.sce, made by Denis Mottet.

Then it loops through the folders in the DAT directory. Each folder corresponds to a set performed at a given speed (slow, average or fast); the folder is named after this speed.

For each set, main.sce creates the path to read the different .csv files needed to perform the analysis then run 3 scripts, in order:

- SolidPosition.sce
- ImuAccelerationRotation.sce
- ComparisonViconImu.sce

They are contained in the PRG directory, in the folder named "executable_scripts"

SolidPosition.sce

This script loads the position data acquired with the Vicon for the three markers X, Y and O; then it estimates the position for the whole IMU.

First it reads the file, then it suppresses the useless information from the matrix created.

The x, y and z coordinates of the solid is estimated from the mean of the x, y and z values respectively of the X and Y markers. Given that they are positioned on opposite sides of the IMU, the mean of their coordinates corresponds to the equidistant point on the imaginary line between them, corresponding to the middle of the IMU.

Finally, it filters the position values to avoid the high frequency noise. We will derivate the signal twice afterwards, we want it to be as clean as possible.

ImuAccelerationRotation.sce

This script loads the acceleration and rotation data from the K-Move and rotates it in the earth frame.

First, it loads the raw data, converts the acceleration data in m.s^{-2} and the gyroscope data in m.s^{-1}, and filter them.

The gyroscope data are then integrated to get the angular values on the x, y and z axis in the IMU referential (also called body frame, noted x^{n}, y^{n} and z^{n} on the ***Figure* 1**).

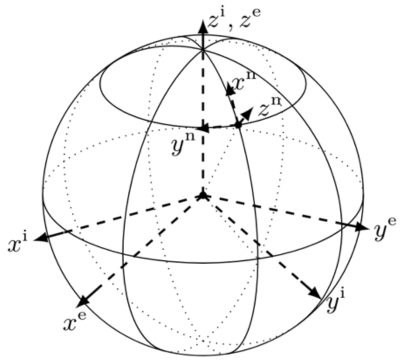


Figure 1 | Representation of the different referential: the n-frame is the body frame, the e-frame is the earth frame

The main role of this script is to use the angular data to rotate the accelerations a_x^n, a_y^n and a_y^n in the earth referential. This way, both acceleration data from the IMU and position data from the Vicon will be in the same referential. To do that, we create a rotation matrix, calculated as:

$$R = R_x \cdot R_y \cdot R_z$$

Where:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_\mathrm{y} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_\mathrm{z} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ComparisonViconImu.sce

This script compares the acceleration data from the Vicon and from the IMU.

First, it synchronizes the signal thanks to the data from the Delsys. Thanks to this file, we detect the onset of the IMU acquisition and cut the position signal from the Vicon.

Next, the position data are derivated twice to obtain acceleration and last, we compare both acceleration data from the IMU and from the Vicon.

3.2. Functions

The functions from the folder "organization_function" are the following:

- **printcenteredmessage**.sci : prints a centered message in the console
- **saveFigure** : saves figure as png file

The functions from the folder "signal_treatment" are the following:

- **fastfft** : create useful fata from a fast Fourier transform operation
- **fltsflts** : filters a signal using a filter with dual pass
- **lowpassbuttdouble** : low pass filter of a signal with dual pass Butterworth.