Relative Translation and Rotation Calibration Between Optical Target and Inertial Measurement Unit

Manthan Pancholi[1(✉)], Svilen Dimitrov[1], Norbert Schmitz[1], Sebastian Lampe[2], and Didier Stricker[1]

[1] German Research Center for Artificial Intelligence, Trippstadter Strasse 122, 67663 Kaiserslautern, Germany
{manthan.pancholi,svilen.dimitrov,norbert.schmitz, didier.stricker}@dfki.de
[2] Volkswagen Group Research, 38436 Wolfsburg, Germany
http://av.dfki.de/

Abstract. Cameras and Inertial Measurement Units are widely used for motion tracking and general activity recognition. Sensor fusion techniques, which employ both Vision- and IMU-based tracking, rely on their precise synchronization in time and relative pose calibration. In this work, we propose a novel technique for solving both time and relative pose calibration between an optical target (OT) and an inertial measurement unit (IMU). The optical tracking system gathers $6DoF$ position and rotation data of the OT and the proposed approach uses them to simulate accelerometer and gyroscope readings to compare them against real ones recorded from the IMU. Convergence into the desired result of relative pose calibration is achieved using the adaptive genetic algorithm.

Keywords: Relative pose calibration · Inertial measurement unit · Tracking calibration · Genetic algorithm

1 Introduction

This paper briefly introduces the reader to the needs in the field of pose tracking for augmented reality applications. Then it summarizes its main contribution of relative pose and time calibration between an OT and an IMU. A survey of the related work is presented before explaining the theoretical part of the developed calibration approach. The largest portion of the paper is the actual implementation, followed by convergence results of the rotation and translation, as well as the time synchronization.

1.1 Motivation

$6DoF$ pose estimation is important in the field of activity recognition, motion recognition, and robotics. Most techniques use visual-based tracking, which suffer

M. Magno et al. (Eds.): S-Cube 2016, LNICST 205, pp. 175–186, 2017.
DOI: 10.1007/978-3-319-61563-9_15

from bad lighting conditions. Other techniques employ IMUs, which suffer from magnetometer distortions and integration drift. Recently, sensor fusion techniques which combine both types of sensors to gain precise tracking, are trending in research. To improve their recognition accuracy they rely on relative pose calibration between the sensors and synchronization in time.

1.2 Contribution

In this work, we propose a novel technique for estimating the three-dimensional translation and three-dimensional rotation offset between an OT and an IMU, as well as synchronization of their readings in time. It first gathers simultaneous data from tracked OT with $6DoF$ pose, and accelerometer and gyroscope readings from IMU over a short period of time. Then it generates simulated accelerometer and gyroscope readings based on the real measurement from the optical tracking system. These simulated readings are compared against the real IMU readings in a genetic algorithm to find out the best fitting $6DoF$ translational and rotational offset between them. Then it repeats the genetic algorithm with shifted readings in order to minimize their time delay. Hardware-wise, besides the optical tracking system and the IMU itself we do not rely on any other external hardware like a turntable or other complex apparatus.

1.3 Related Work

With the research in vision- and inertial-based tracking and their simultaneous usage emerges the need of their calibration as a combination. J. Alves and J. Lobo published first results in this field by aligning the rotation between a camera and an IMU using vertical vision features and the vertical gravity vector measured by an accelerometer [1]. They later refined their work by including the translation calibration using a simple passive turntable and static images [6]. A year later Mirzaei and Roumeliotis proposed a Kalman filter-based algorithm for IMU-camera calibration, where they removed the calibration constraints of a special setup [8]. The follow-up research continued refining those results by using different Kalman filter adaptations and hardware [4,5,10]. In filtering framework approach, estimation of the pose of IMU is required, which is generally non-trivial to achieve and requires complex modeling. Yet all of the mentioned research considers a relative pose calibration between a camera and an IMU, while in our case we are interested in the relative pose calibration between an OT and an IMU using the adaptive genetic algorithm in corporate with simulating the IMU readings. In this sense, our research direction is also similar to the estimation of the relative 6DoF pose. To our knowledge, the relative pose calibration by simulating IMU readings is not researched yet.

2 Relative Pose Calibration Approach

In this section, the calibration procedure is formulated to determine $6DoF$ relative pose between the OT and the IMU. We start by introducing three separate reference frames which have been considered.

- *Global frame G:* The pose of the OT is represented with respect to this coordinate frame which is fixed in the environment.
- *Optical frame O:* The OT represents this frame.
- *IMU frame I:* The IMU represents this frame and all the inertial measurements are expressed in this coordinate frame.

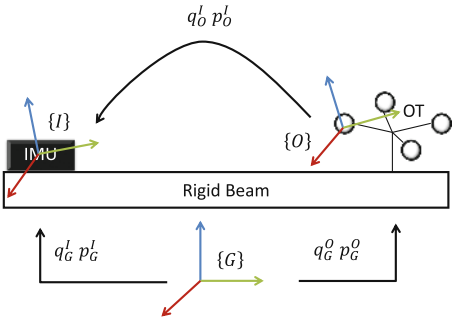

Fig. 1. Global (G), Optical (O) and IMU (I) reference frames (x-axis red, y-axis green, z-axis blue). The unknown transformation from the OT frame to the IMU reference frame can be expressed by unit quaternion q_O^I (rotation from O to reference frame I) and translation vector P_O^I (3D position of the IMU with respect to frame O). Similarly, the known pose of the OT is expressed by q_G^O (rotation from G to O) and P_G^O (3D position of the OT with respect to frame G). (Color figure online)

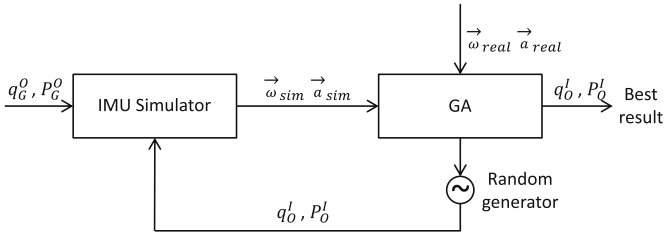

Fig. 2. Calibration algorithm flow diagram

Our approach consists of two major schemes, namely *IMU Simulator* and *Genetic Algorithm (GA)*. Figure 2 illustrates overview of our calibration algorithm. The IMU Simulator uses the recorded tracking data (q_G^O, P_G^O) of the OT and the unknown offsets (q_O^I, P_O^I) randomly generated by the GA to simulate corresponding IMU readings, represented as simulated angular velocity $\vec{\omega}^I_{sim}$ and acceleration $\vec{a}^I_{sim}$ in the IMU reference frame. As a next step, simulated data is compared with the real data captured from the IMU to calculate the fitness values as explained later in Subsect. 2.2. Depending on the fitness values, GA generates new offsets in each iteration until the best individual (q_O^I, P_O^I) is found. To demonstrate the ability of our proposed approach, methodology has been discussed in the following sections.

2.1 Inertial Sensor System Simulation

Using optical tracking system, we can perceive "tracking" as measurement of the position and orientation (q_G^O, P_G^O) of the OT with multiple markers. Inertial sensor system simulation uses these measurements to simulate IMU readings in terms of angular velocity and acceleration. The IMU and the OT are fixed on a single rigid body as shown in Fig. 1. Generally, both these system data arrives at different frequency rates. Therefore, it is necessary to compute the pose of the OT at the same rate as the arriving IMU data for comparison. To solve this problem we interpolate the q_G^O and P_G^O at the timestamps t^{IMU} of real IMU. There are several methods available to interpolate the 3D positions and unit quaternion orientations. Interpolating in quaternion space using unit quaternion representation ensures a unique path under all circumstances. For the position interpolation we are using *Cubic Spline Interpolation* method with a "not-a-knot" condition, which means that, at the first and last interior break, even the third derivative is continuous (up to round-off error). So, there are no breaks at any knots. Continuity of this method is C^2. In our case we have known P_G^O positions (knots) corresponding to optical timestamps $t^{optical}$. So, for the given data points ($t_1^{optical}$, $P_{G,1}^O$), ..., ($t_n^{optical}$, $P_{G,n}^O$), $S(t^{optical})$ is a cubic interpolating spline function for this data if [7],

$$S(t_i^{optical}) = P_{G,i}^O, i = 1, ..., n. \tag{1}$$

Applying "not-a-knot" condition, which is:

$$\dddot{S}_1(t_2^{optical}) = \dddot{S}_2(t_2^{optical}), \dddot{S}_{n-2}(t_{n-1}^{optical}) = \dddot{S}_{n-1}(t_{n-1}^{optical}), \tag{2}$$

Spherical Linear Interpolation (SLERP) method can be used to interpolate between two unit quaternions. But, while interpolating between series of the unit quaternions, this method doesn't provide smooth interpolation curve at the nodes (q_G^O at $t^{optical}$). To interpolate unit quaternion rotations of the OT at t^{IMU} with C^2 continuity in interpolation curve, *Spherical Spline Quaternion Interpolation (SQUAD)* [2] is used. The *SQUAD* method does cubic interpolation between data points $q_{G,i}^O$ and $q_{G,i+1}^O$ using Eqs. 3 to 7.

$$Slerp(q_{G,i}^O, q_{G,i+1}^O, T) = q_{G,i}^O \frac{sin((1-T)\theta)}{sin(\theta)} + q_{G,i+1}^O \frac{sin(T\theta)}{sin(\theta)} \tag{3}$$

$$cos(\theta) = q_{G,i}^O \cdot q_{G,i+1}^O \tag{4}$$

Where $T \in [0, 1]$ depending on the value of t^{IMU} lying between two consecutive $t_i^{optical}$ and $t_{i+1}^{optical}$ where we want to interpolate the quaternion,

$$T = \frac{t^{IMU} - t_i^{optical}}{t_{i+1}^{optical} - t_i^{optical}}, \tag{5}$$

$$\begin{aligned} Squad(q_{G,i}^O, & q_{G,i+1}^O, s_i, s_{i+1}, T) \\ & = Slerp(Slerp(q_{G,i}^O, q_{G,i+1}^O, T), Slerp(s_i, s_{i+1}, T), 2T(1-T)), \end{aligned} \tag{6}$$

The point s_i and s_{i+1} are called inner quadrangle points which guarantee continuity across segments. For the data set of unit quaternions $(q^O_{G,1}, q^O_{G,2}, ..., q^O_{G,n})$, $s_1 = q^O_{G,1}$ and $s_n = q^O_{G,n}$.

$$s_i = q^O_{G,i} \exp(-\frac{\log((q^O_{G,i})^{-1} * q^O_{G,i+1}) + \log((q^O_{G,i})^{-1} * q^O_{G,i-1})}{4}) \tag{7}$$

After applying interpolation procedures, the IMU pose with respect to the global frame is calculated using Eqs. 8 and 9, by considering unknown relative pose (q^I_O, P^I_O) and known pose (q^O_G, P^O_G) at t^{IMU} timestamps.

$$q^I_G = q^O_G q^I_O, \tag{8}$$

$$P^I_G = P^O_G + q^O_G P^I_O conj(q^O_G), \tag{9}$$

The reason behind interpolating the pose corresponding to timestamps t^{IMU} is to calculate simulated instantaneous inertial measurement readings to compare with real readings acquired from the IMU at timestamps t^{IMU}. The simulated angular velocity $\overrightarrow{\omega}^I_{sim}$ in the IMU frame is calculated by taking the derivative of quaternion $\frac{dq}{dt}$ as expressed in Eq. 10.

$$\overrightarrow{\omega}^I_{sim} = 2conj(q^I_G)\dot{q}^I_G, \tag{10}$$

An accelerometer measures the external specific force acting on the IMU sensor. The specific force consists of both, the sensor's acceleration and the Earth's gravity. Also, the IMU measures gravitational acceleration in the opposite direction of gravitational force. In our case gravity is acting towards the negative Z-axis direction of global reference frame (G). So, the gravitational acceleration vector $\overrightarrow{a}^G_g$ consists positive acceleration g in Z-axis direction (Eq. 11).

$$\overrightarrow{a}^G_g = (0, 0, 9.81) ms^{-2}, \tag{11}$$

Acceleration in G frame $\overrightarrow{a}^G_{I,i}$ is calculated by adding $\overrightarrow{a}^G_g$ to the net linear acceleration from the change in positions P^I_G at each time instances t^{IMU} according to Eq. 12. But, to compare with real IMU accelerometer readings, $\overrightarrow{a}^G_{I,i}$ needs to be expressed in the IMU reference frame by rotating it according to q^I_G as in Eq. 13.

$$\overrightarrow{a}^G_{I,i} = \frac{v^G_i - v^G_{i-1}}{t^{IMU}_i - t^{IMU}_{i-1}} + \overrightarrow{a}^G_g, \tag{12}$$

$$\overrightarrow{a}^I_{sim,i} = conj(q^I_{G,i})\overrightarrow{a}^G_{I,i} q^I_{G,i}, \tag{13}$$

Where, v^G_i is the velocity at i^{th} time instance t^{IMU}_i. Calculated $\overrightarrow{\omega}^I_{sim}$ and $\overrightarrow{a}^I_{sim}$ will be compared with the real IMU data $(\overrightarrow{\omega}^I_{real}, \overrightarrow{a}^I_{real})$ in *Genetic Algorithm* technique to solve the calibration problem.

2.2 Genetic Algorithm

In this work, we propose the usage of an Adaptive Genetic Algorithm [3] to find the rotational and positional offsets between the OT and the IMU. It maintains a population of n possible solutions with associated fitness values. Parents, in this case rotations or translations, are sorted to produce new population based on their fitness value, being the mean value of the difference between the simulated values ($\overrightarrow{\omega}^I_{sim}$, $\overrightarrow{a}^I_{sim}$) and the real values ($\overrightarrow{\omega}^I_{real}$, $\overrightarrow{a}^I_{real}$), explained later in this section. New generations of solutions are produced near the top previous solutions using uniform distribution, which contain on average more good genes than previous generation. Once the population has converged and is not producing new populations noticeably different from those in previous generations, the algorithm itself is said to have converged to a set of solutions to the problem at hand. This concept should be applied two times, first to get rotation offset q^I_O while keeping translation offset $P^I_O = 0$ followed by the second step, which involves fixing the rotation offset obtained in the first part and searching for the best translation offset P^I_O. Note that the above is possible since we can use a gyroscope simulation readings for comparison, which should not be affected by the translation. In this case, fitness values F_{gyro} are calculated according to the Eq. 14.

$$F_{gyro} = \frac{\sum_{i=1}^{N} |\overrightarrow{\omega}^I_{sim,i} - \overrightarrow{\omega}^I_{real,i}|}{N}, \tag{14}$$

For the translational offset approximation, the accelerometer readings are used to calculate fitness values F_{acclr}. Where N is a number of samples.

$$F_{acclr} = \frac{\sum_{i=1}^{N} |\overrightarrow{a}^I_{sim,i} - \overrightarrow{a}^I_{real,i}|}{N}, \tag{15}$$

3 Experiment

3.1 Hardware

To justify the accuracy of our calibration algorithm we carried out several experiments with hardware including, *ART* optical tracking system and the *EPSON* IMU. *ART* provides $6DoF$ pose (q^O_G, P^O_G) of the optical target with four markers, these measurements are retrieved through *DTrack2 SDK* at 60 Hz frequency. The IMU is a M-G350-PD11 model manufactured by *EPSON*, which provides angular velocity $\overrightarrow{\omega}^I_{real}$ and acceleration data $\overrightarrow{a}^I_{real}$ at 125 Hz. Both of these are mounted on a 45 cm long rigid beam (see Fig. 3), the IMU on one end and the OT on the other end with freedom to vary the distance in between them.

Fig. 3. 45 cm long rigid beam mounted with OT and IMU

3.2 Data Acquisition

At the start of each experimental trial we perform free non-specified movement around each axis of the beam for approximately 30 s, and gather the data from OT and the IMU system. There is no constraint on how one should move the beam during experiment. $6DoF$ pose is interpolated from 60 Hz to 125 Hz as explained in Sect. 2.1 to get $\overrightarrow{\omega}^I_{sim}$ and $\overrightarrow{a}^I_{sim}$. Random generation of q^I_O and P^I_O is explained in Sect. 3.3.

3.3 Genetic Algorithm Implementation and Parameters

Algorithm 1 illustrates a pseudo-code of the implementation of the genetic algorithm:

Initial Run: Using uniform distribution, here we generate 10000 random seeds (q^I_O, P^I_O), out of which 1000 best are selected for our adaptation iterations. We place no limit on the rotation direction, while for the translation all the three axes are limited to be in the range of $[-1, 1]$ meters. In order to evaluate our resulted genes (q^I_O, P^I_O) we compute corresponding simulated gyroscope and acceleration readings for comparison with the real ones by calculating the average over the values of the vector difference between simulated and real vectors (Eqs. 13 and 14).

Since the double differentiation of discrete positions generates artifacts, window based sliding average smoothing over the simulated acceleration readings is required before comparing them with the real ones [9]. We considered a window size of 7 discrete values.

Input: 60 Hz 6DoF optical target data OTD consisting of 3-D position $OTPos$ and 3-D rotation $OTRot$, 125 Hz 3-D accelerometer $AReal$ and 3-D gyroscope $GReal$ inertial measurement unit data $IMUD$
Output: 6-DoF translational v and rotational r offset between the OT and the IMU as well as their time synchronization t

for $s \leftarrow -10$ ***to*** 10 **do**
  shift OTD by s
  synchronize OTD rate to $IMUD$ rate with Spline and SQUAD

  $v_s \leftarrow (0, 0, 0)$
  for $i \leftarrow 0$ ***to*** 9999 **do**
    $rq_i \leftarrow$ random quaternion
    $GSim \leftarrow$ simulated values according to rq and v_s
    $qdist_i \leftarrow avg(|GSim - GReal|)$
  $sort(rq$ by $qdist)$
  $maxangle \leftarrow 60$
  for $repeat \leftarrow 0$ ***to*** 9 **do**
    for $i \leftarrow 0$ ***to*** 249 **do**
      for $axis \leftarrow 0$ ***to*** 2 **do**
        $maxangle \leftarrow maxangle * (repeat - 1)/repeat$
        $rangle \leftarrow$random angle between $\pm maxangle$
        $rq[250 + i * axis] \leftarrow rq_i$ rotated over $axis$ by $rangle$
        $GSim \leftarrow$ simulated angular velocities according to rq
        $qdist_i \leftarrow avg(|GSim - GReal|)$
    $sort(rq$ by $qdist)$

  $r_s \leftarrow rq_0$
  $maxdist \leftarrow \pm 1$
  for $i \leftarrow 0$ ***to*** 9999 **do**
    $rv_i \leftarrow$ random vector; $ASim \leftarrow$ simulated values according to rv and q_s
    $vdist_i \leftarrow avg(|ASim - AReal|)$
  $sort(rv$ by $vdist)$
  $maxdist \leftarrow 0.1$
  for $repeat \leftarrow 0$ ***to*** 9 **do**
    for $i \leftarrow 0$ ***to*** 249 **do**
      for $axis \leftarrow 0$ ***to*** 2 **do**
        $maxdist \leftarrow maxdist * (repeat - 1)/repeat$
        $vtrans \leftarrow$random vector between $\pm maxdist$
        $rv[250 + i * axis] \leftarrow rv_i$ translated over $axis$ by $vtrans$
        $ASim \leftarrow$ simulated accelerations according to rv
        $smooth(ASim)$
        $vdist_i \leftarrow avg(|ASim - AReal|)$
    $sort(rv$ by $vdist)$
  $v_s \leftarrow rv_0$
$t \leftarrow s[min(v_s + q_s)]$
return t, v_s, r_s

Algorithm 1. CALIBRATE-OT-IMU synchronizes and calibrates the relative translation and orientation between optical target and inertial measurement unit

Adaptation Iterations: We converge to the desired rotation or translation by filtering out the best 250 results out of 1000. Then for each axis we rotate or displace by a given factor f, depending on the iteration progress. The factor is being reduced according to the formula in Eq. 16.

$$f = \frac{iterations - 1}{iterations} \tag{16}$$

The initial rotation range is set in the range $\pm 60°$, while the translation range is set to ± 10 cm. 10 iterations are performed, which means that on the second iteration the rotational range will be $\pm 54°$ and the translation range will be ± 9 cm. We choose again a random value in the range using uniform distribution. This way, 250 best population from the previous iteration are stored and 750 new population around them are generated in each iteration.

Time Shifting: We store the best result and its fitness value after each run of the genetic algorithm. Then, the tracked IMU real data is shifted over the time line followed by rerunning the algorithm to see for which time shift the best result is achieved. We do this, because our initial synchronization suffers from hardware constraints like IMU data latencies.

4 Results

The effectiveness of the proposed approach is measured in terms of how it converges to the desired results in terms of rotation offset, translation offset, and time synchronization. The translation reference estimate (T) for this experiment is obtained by measuring the distance of the IMU reference frame I origin from the optical target frame O origin based on the information provided in the data sheets of *ART*[1] and *EPSON* IMU[2]. During the experiments, the translation offset from O to I is $T = (0.40, 0.025, -0.07)$ m and the rotation offset was set to be zero using calibration technique provided in *ART* tracking system. By plotting the data captured from both sensor systems, time delay was measured to be between 32 and 40 ms. An exact time delay during the experiment couldn't be measured due to sampling rate disturbances in both DTrack and Epson recording soft- and hardware. We performed several experiments with this ground truth and ran the algorithm multiple times to measure its different outputs. From those the computed translations had an average 3D difference of $(0.0032837, 0.0059151, -0.00427)$ m, and the quaternion rotation had component wise an average difference of $(0.000656, -0.00515105, -0.00317474, 0.0357037)$. Sections 4.1, 4.2 and 4.3 show our genetic algorithm convergence to its best chosen rotation offset, translation offset, and time offset in the experiment with the mentioned reference estimates.

[1] ART System User Manual, version 2.1, April 2015.
[2] M-G350-PD11 Datasheet, 21 October 2012.

4.1 Rotation Convergence

Figure 4 illustrates 3 different iteration steps of our rotational convergence:

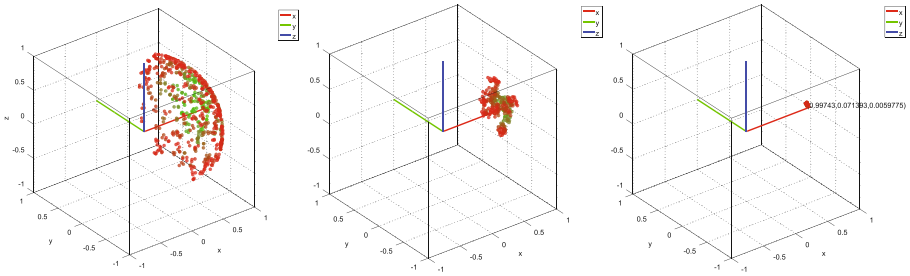

Fig. 4. 3-D rotational convergence to (1, 0, 0, 0) (Color figure online)

In (a) we see the 1000 best results after filtering with sorting the initial seed phase consisting of 10000 seeds. In (b) we see iteration 3 and in (c) is iteration 10, our best result which belongs to quaternion $q_O^I = (0.999344, -0.00515105, -0.00317474, 0.0357037)$, and is close to the rotation reference estimate $q_{O,ref.}^I = (1, 0, 0, 0)$. The dots represent rotations of vector (1, 0, 0) according to the generated quaternion seeds. And the color transition from red to green represents bad and good fitness values respectively. Which means best obtained q_O^I will rotate vector (1, 0, 0) to new position (0.99743, 0.071393, 0.0059775) as in Fig. (c).

4.2 Translation Convergence

Figure 5 illustrates 3 different iteration steps of our translational convergence:

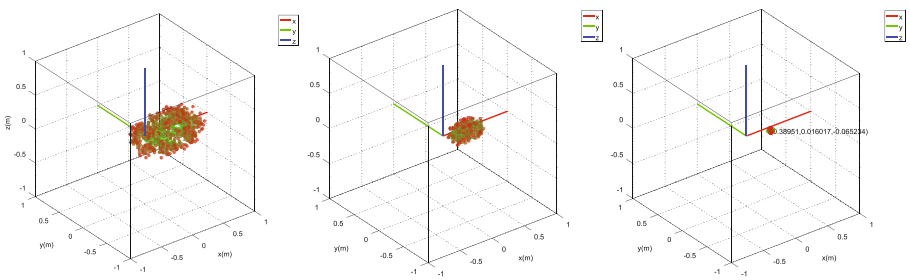

Fig. 5. 3-D translational convergence to $(0.40, 0.025, -0.07)$ m (Color figure online)

In (a) we see the 1000 best sorted results after filtering the initial seed phase consisting of 10000 seeds. In (b) we see iteration 3 and in (c) iteration 10, our best result which belongs to translation $P_O^I = (0.389506, 0.0160174, -0.0652339)$ m, and is closer to the translation reference estimate $T = (0.40, 0.025, -0.07)$ m. The color transition from red to green represents bad and good fitness values respectively.

4.3 Time Synchronization

For the obtained best results q_O^I and P_O^I, our best time match varied between 40 and 32 ms, which corresponds to five and four samples arriving at 125 Hz rate respectively. This has been justified by Figs. 6 and 7 representing the comparison between real and simulated values of the Gyroscope and Accelerometer.

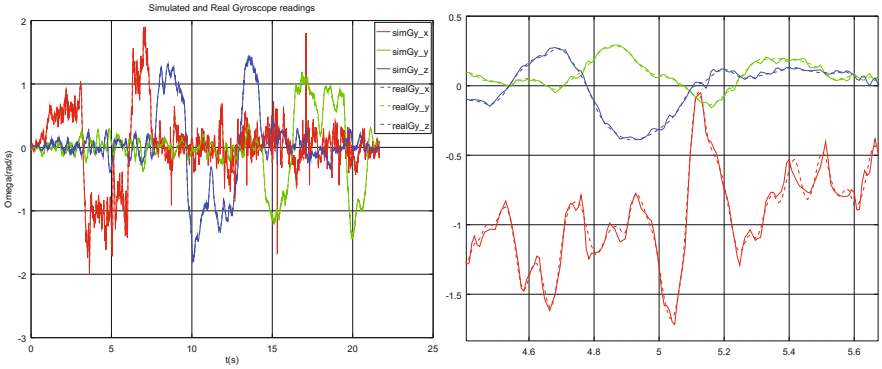

Fig. 6. Time synchronization comparison between simulated (solid) and real (dotted) accelerometer readings. Left full experiment plot, right is the magnified portion of the same plot.

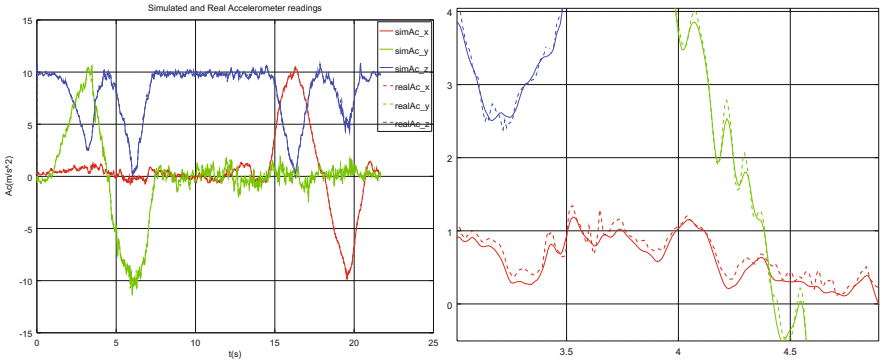

Fig. 7. Time synchronization comparison between simulated (solid) and real (dotted) gyroscope (right) readings. Left full experiment plot, right is the magnified portion of the same plot.

Figures 6 and 7 indicate that both simulated and real accelerometer, gyroscope readings are significantly close to each other. Which justifies the resulted best orientation offset q_O^I and the translation offset P_O^I obtained in this experiment.

5 Conclusion

This paper described a novel approach to accurately determine the 3D rotational offset, 3D translational offset and a time synchronization in data between an optical target and an inertial measurement unit. It consists of creating simulated acceleration and gyroscope readings by applying 6-DoF offset to the optical target and compare them against the real ones from the IMU in an adaptive genetic algorithm. The experiments shown in the previous section suggest that the developed system delivers admirable results in all three tasks it has been designed for. Furthermore, it is not dependent on supplementary hardware or other constraints besides the factory calibration of the optical tracking system and its target as well as the inertial measurement unit calibration. Additionally, for operation in different environment, our method enables rapid re-calibration when the relative pose of the sensors must be changed. Thus, we consider the technique as applicable in the fields of augmented reality, robotics and automotive industry. We plan to continue our work in two directions - by making more extensive evaluations of our system and implementing a sensor fusion techniques, which relies on it.

References

1. Alves, J., Lobo, J., Dias, J.: Camera-inertial sensor modelling and alignment for visual navigation. Mach. Intell. Robotic Control **5**(3), 103–112 (2003)
2. Dam, E.B., Koch, M., Lillholm, M.: Quaternions, Interpolation and Animation. Københavns Universitet, Datalogisk Institut (1998)
3. Davis, L.: Handbook of Genetic Algorithms, 1st edn. Van Nostrand Reinhold, New York (1991)
4. Hol, J.D., Schön, T.B., Gustafsson, F.: Modeling and calibration of inertial and vision sensors. Int. J. Robotics Res. **29**(2–3), 231–244 (2010)
5. Kelly, J., Sukhatme, G.S.: Fast relative pose calibration for visual and inertial sensors. In: Khatib, O., Kumar, V., Pappas, G.J. (eds.) Experimental Robotics, vol. 54, pp. 515–524. Springer, Heidelberg (2009)
6. Lobo, J., Dias, J.: Relative pose calibration between visual and inertial sensors. Int. J. Robotics Res. **26**(6), 561–575 (2007)
7. McKinley, S., Levine, M.: Cubic spline interpolation. College Redwoods **45**(1), 1049–1060 (1998)
8. Mirzaei, F.M., Roumeliotis, S.I.: A kalman filter-based algorithm for IMU-camera calibration: observability analysis and performance evaluation. IEEE Trans. Robotics **24**(5), 1143–1156 (2008)
9. O'Haver, T.: A Pragmatic Introduction to Signal Processing. CreateSpace Independent Publishing Platform, North Charleston (1997)
10. Zheng, S., Chai, X., Su, S., Liu, X., Neta, K., Miller, W.: Relative pose calibration between inertial unit and visual unit in railway track inspection system. J. Balk. Tribological Assoc. **22**(2), 1253–1264 (2016)