Human Arm Motion Tracking by Kinect Sensor Using Kalman Filter for Collaborative Robotics

Human Arm Motion Tracking by Kinect Sensor using Kalman Filter for Collaborative Robotics

Pierpaolo Palmieri[1], Matteo Melchiorre[1], Leonardo Sabatino Scimmi[1], Stefano Pastorelli[1] and Stefano Mauro[1]

[1] Department of Mechanical and Aerospace Engineering, Politecnico di Torino, C.so Duca degli Abruzzi 24, 10129, Torino, Italy
{pierpaolo.palmieri, matteo.melchiorre, leonardo.scimmi, stefano.pastorelli, stefano.mauro}@polito.it

Abstract. The rising interest in collaborative robotics leads to research solutions in order to increase robot interaction with the environment. The development of methods that permit robots to recognize and track human motion is relevant for safety and collaboration matters. A large quantity of data can be measured in real time by Microsoft Kinect®, a well-known low-cost depth sensor, able to recognize human presence and to provide postural information by extrapolating a skeleton. However, the Kinect sensor tracks motion with relatively low accuracy and jerky behavior. For this reason, the effective use in industrial applications in which the measurement of arm velocity is required can be unsuitable. The present work proposes a filtering method that allows the measurement of more accurate velocity values of human arm, based on row data provided by the Kinect sensor. The estimation of arm motion is achieved by a Kalman filter based on a kinematic model and by the imposition of fixed lengths for the skeleton links detected by the sensor. The development of the method is supported by experimental tests. The achieved results suggest the practical applicability of the developed algorithms.

Keywords: Kalman filter, collaborative robotics, Microsoft Kinect, human motion.

1 Introduction

Collaborative robotics is a topic of rising interest in industrial applications. The human-robot interaction (HRI) combines the advantages of traditional industrial robots such as speed, strength and accuracy with human dexterity, flexibility and intelligence. In a HRI system, robots and humans jointly work on the same tasks improving on efficiency and productivity and reducing the human workload [1]. Since humans and robots can share the same workspace, a special attention must be paid on safety [2].

Different kinds of sensors are suitable to track human motion for HRI. Wearable sensors, e.g. inertial measurement units, have low computational workload, but they are uncomfortable for the user. On the other hand, non-wearable sensors, e.g. depth sensors, avoid contact with human body, but have low accuracy [3]. Microsoft Kinect is a low-cost-device with widespread use in computer vision [4]. It is a depth camera equipped with RGB sensor and microphone. In HRI systems, it can be used to perform gesture recognition for robot motion control [5] or to monitor the environment and detect humans [6], for instance, allowing robots to execute flexible and free collision trajectories for safer operation on dynamic workspaces [7].

By using Kinect and its Software Development Kit (SDK) [8], it is possible to track the 3D human posture. The main outputs are a point cloud, representing a 3D view of the surrounding environment including people, and a skeleton representation of people in the view field, with 3D joint position information, in a reference frame fixed with the camera. Kinect version 2 can work with a frequency up to 30 Hz and can detect 25 skeleton joints with a maximum of 6 skeletons, corresponding to 6 people, at the same time.

However, the skeleton raw signal provided by Kinect is not always suitable for industrial collaborative robotics, due to low accuracy, movement discontinuity and variation in the link length [9]. In order to overcome this limitation, the point cloud can be used as reference of the human position [10]. Nevertheless, the point cloud is harder to manage due to the larger amount of data and it requires longer computational time. An alternative approach is about the enhancement of the skeleton signal through a Kalman filter. The use of a Kalman filter can produce data that tend to be closer to the true spatial measurements, starting from signals, with noises and uncertainties, captured by one or more sensors. For these reasons, it has become a popular approach in robot vision applications [11]. Studies have shown that the Kalman filter has a best overall performance compared to other filter-based approaches for Kinect skeleton noise reduction [12].

In this paper a new method is proposed to achieve more suitable data for human arm tracking by the Kinect skeleton. A Kalman filter is applied to joint variables, calculated through a human arm model that considers kinematic constraints of body movements. Moreover, a fixed link length approach is defined to reduce the noises of the skeleton joints. The proposed methods have been developed to increase the performance of collision avoidance and hand over algorithms described in [13-15]. They have the potentiality to be applied using different models and extended to wider human body regions.

2 Methodology

The method hereby described intends to process data measured by Kinect in order to filter the effect of noise, temporary occlusions and measurement errors. Among the skeleton joints provided by Kinect, five joints have been used in this work, as shown in Fig.1 (a).

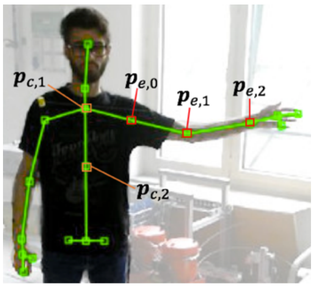
The human arm posture is described by the position vectors of the joints shoulder $\boldsymbol{p}_{e,0}$, elbow $\boldsymbol{p}_{e,1}$ and wrist $\boldsymbol{p}_{e,2}$. In order to complete the algorithm, two joints that belong to the chest have been selected: a point between the left and right shoulder $\boldsymbol{p}_{c,1}$ and a point in the middle of the spine $\boldsymbol{p}_{c,2}$.

The present dissertation focuses on the analysis of the left arm motion, nonetheless it is valid for both the arms.

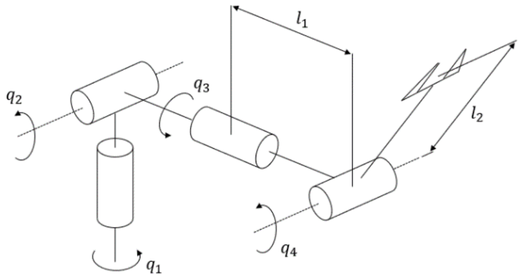
2.1 Human Arm Model

The human arm is modeled with two links, arm and forearm, with four degrees of freedom, three for the shoulder and one for the elbow, as schematized in Fig. 1 (b). Then, four revolute joint variables have been identified and collected in the vector of joint variables $\boldsymbol{q}$. The configuration with all the joint variables at zero value corresponds to the posture with arm extended and raised above the shoulder.

Through the forward kinematics (fkine) it is possible to obtain the position vectors from the joint variables and the link lengths l_1 and l_2. A closed-form solution for the inverse kinematics (ikine) problem is found considering the constraints $-\pi \leq q_2 \leq 0$ and $0 \leq q_4 \leq \pi$, related to a natural human arm motion.



(a)



(b)

Fig. 1. Kinect v2 upper body skeleton joints (a) and human arm kinematic model (b)

2.2 Kalman filter

A Kalman filter [16] is used to reduce signal noises referred to the joint space variables. A space-state model for the process is required:

$$\boldsymbol{x}_{k+1} = \boldsymbol{G}_k \boldsymbol{x}_k + \boldsymbol{w}_k \tag{1}$$

$$\boldsymbol{z}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k \tag{2}$$

where $\boldsymbol{x}$ is the state vector, $\boldsymbol{G}$ is the state transition matrix, $\boldsymbol{w}$ is the process noise, $\boldsymbol{z}$ is a measurement vector, $\boldsymbol{H}$ is the measurement matrix, and **v** is the measurement noise at the instant k. The algorithm for a Kalman filter consists of an iterative prediction-update process. At each iteration, the state and its covariance are predicted from the previous state with the use of a model by the state transition matrix. Then, the measurement is used to calculate a more accurate state and covariance, updating the predicted values. The algorithm is described by the following equations:

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_K)^{-1} \tag{3}$$

$$\boldsymbol{x}_k^+ = \boldsymbol{x}_k^- + \boldsymbol{K}_k (\boldsymbol{z}_k - \boldsymbol{H}_k \boldsymbol{x}_k^-) \tag{4}$$

$$\boldsymbol{P}_k^+ = \boldsymbol{P}_k^- + \boldsymbol{K}_k \boldsymbol{H}_k \boldsymbol{P}_k^- \tag{5}$$

$$\boldsymbol{x}_{k+1}^- = \boldsymbol{G}_k \boldsymbol{x}_k^+ \tag{6}$$

$$\boldsymbol{P}_{k+1}^{-} = \boldsymbol{G}_k \boldsymbol{P}_k^{+} \boldsymbol{G}_k^{T} + \boldsymbol{Q}_k \tag{7}$$

where $\boldsymbol{K}$ is the Kalman gain, $\boldsymbol{x}^{-}$ is the a priori state estimate, $\boldsymbol{P}^{-}$ is the a priori estimate error covariance, $\boldsymbol{x}^{+}$is the a posteriori state estimate, $\boldsymbol{P}^{+}$ is the a posteriori estimate error covariance, $\boldsymbol{Q}$ is the covariance matrix of the process noise and $\boldsymbol{R}$ is the covariance matrix of the measurement noise.

A Wiener Process Acceleration (WPA) state model [17] is applied to the Kalman Filter. It is a null jerk model that describes the system by the following expressions:

$$\boldsymbol{x} = [\hat{\boldsymbol{q}} \quad \hat{\dot{\boldsymbol{q}}} \quad \hat{\ddot{\boldsymbol{q}}}]^T \tag{8}$$

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I} \cdot \Delta t & \boldsymbol{I} \cdot \Delta t^2/2 \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{I} \cdot \Delta t \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{9}$$

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{I} \cdot \Delta t^5/20 & \boldsymbol{I} \cdot \Delta t^4/8 & \boldsymbol{I} \cdot \Delta t^3/6 \\ \boldsymbol{I} \cdot \Delta t^4/8 & \boldsymbol{I} \cdot \Delta t^3/3 & \boldsymbol{I} \cdot \Delta t^2/2 \\ \boldsymbol{I} \cdot \Delta t^3/6 & \boldsymbol{I} \cdot \Delta t^2/2 & \boldsymbol{I} \cdot \Delta t \end{bmatrix} \tilde{\sigma}_q \tag{10}$$

where $\hat{\boldsymbol{q}}$ is the joint variables vector estimation, $\boldsymbol{I}$ is a square identity matrix having the number of joint variables as dimension, Δt is the sample time (for the Kinect sensor Δt = 1/30 s) and $\tilde{\sigma}_q$ is the process noise density. To be thorough it is specified:

$$\boldsymbol{z} = \boldsymbol{q} \tag{11}$$

$$\boldsymbol{H} = [\boldsymbol{I} \quad \boldsymbol{0} \quad \boldsymbol{0}] \tag{12}$$

$$\boldsymbol{R} = \boldsymbol{I} \cdot \sigma_r^2 \tag{13}$$

where $\boldsymbol{q}$ is the raw joint variables vector and σ_r^2 is the variance of the measured noise.

2.3 Shoulder frame

The human arm model has been built considering the shoulder frame, but the position vectors are initially expressed in a fixed frame. The shoulder frame origin is situated in the shoulder joint position and its orientation expresses the chest orientation for each instant. It is calculated using the shoulder joint position $\boldsymbol{p}_{e,0}$ and two additional points $\boldsymbol{p}_{c,1}$ and $\boldsymbol{p}_{c,2}$. A position vector, expressed in the shoulder frame, is written as $\boldsymbol{p}_{e,n}^{O}$.

A low pass single pole IIR filter [18] is applied to the indicated points to cut high frequency displacement, due to the unprocessed raw data noise. The actual motion of these joints typically shows low frequency. Experimental tests show that filtering at 0.2 Hz allows a good tracking.

2.4 Process architecture and link length

The main inputs for the developed algorithm are the arm joints position vectors expressed in a world frame, indicated with $\boldsymbol{p}_{e,n}^{W}$. It is a fixed frame defined during a startup calibration phase. Fig. 2 shows the block diagram of the algorithm, that processes the signal for each instant k.

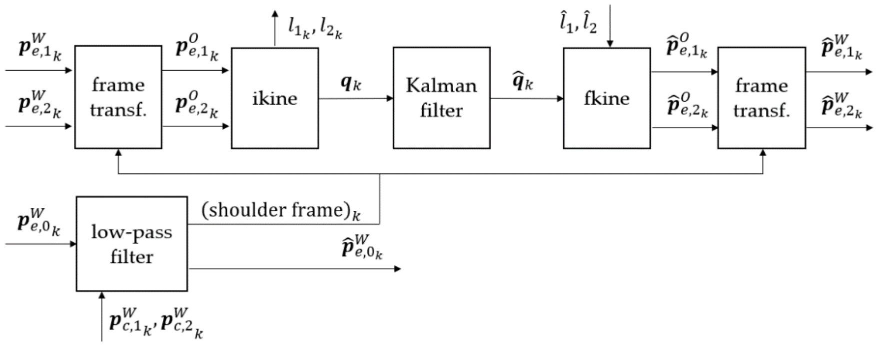


Fig. 2. System block diagram

At first, the low pass filter is applied to obtain the shoulder frame and the filtered shoulder position vector $\hat{\boldsymbol{p}}^W_{e,0}$. Then, the elbow and wrist position vectors are expressed in the shoulder frame, respectively $\boldsymbol{p}^O_{e,1}$and $\boldsymbol{p}^O_{e,2}$. The inverse kinematics is applied and the joint variables vector $\boldsymbol{q}$ and the current link lengths can be calculated. The Kalman filter processes the joint variables using a kinematic model providing a better estimation of them. The position vectors of elbow and wrist, respectively $\hat{\boldsymbol{p}}^O_{e,1}$and $\hat{\boldsymbol{p}}^O_{e,2}$, are estimated applying the forward kinematics algorithm. To achieve better results, the links lengths l_1 and l_2 are assumed constant and equal to values measured or estimated before starting the measurement process. Finally, the estimated position vectors are expressed in the world frame.

3 Experimental Set-up

The experimental set-up consists in a Kinect version 2 sensor and three markers, as shown in Fig. 3 (a). The joint positions provided by Kinect are referred to its own reference frame, called Kinect frame. A set of 3D-printed markers are used to locate a fixed frame, that is assumed as world frame. It is identified in a calibration phase, during which three points are extrapolated from the point cloud using the markers as reference. Finally, the transformation between Kinect and world frame is performed.

In order to evaluate the result of the developed algorithm, a comparison with a vision sensor with better performance is accomplished. The chosen sensor is OptiTrack Trio, a marker-based sensor. It consists of three cameras with 640 x 480 resolution, frame rate up to 120 fps, sub-millimeter accuracy and 8.3 ms latency [19].

Two markers are collocated with the wrist and shoulder of the user. As shown in Fig. 3 (b), it exists an off-set between the joint positions measured by OptiTrack and Kinect because OptiTrack measures the position of the markers, which are close to the arm but external to it, while Kinect measures the position of a virtual joint identified by the analysis of point cloud data. The relative position of the markers detected by OptiTrack and the points detected by Kinect is not constant. This involves that a comparison of the measured position of the corresponding points detected by the two measurement instruments cannot lead to any conclusion. On the contrary, a meaningful comparison can be carried out between the measured speeds of the corresponding points.

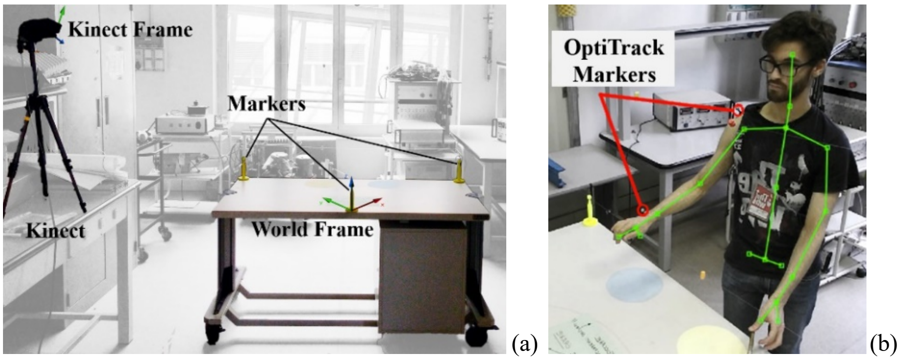


Fig. 3. Experimental set-up. Kinect and markers for world frame identification (a) and OptiTrack markers compared to skeleton joints (b)

4 Results

The outputs of the algorithm are the position vectors of the arm joints. The most interesting joint to analyze is the wrist due to its wider workspace. The results depend on the Kalman filter parameters $\tilde{\sigma}_q$ and σ_r^2, whose optimal values are a-priori unknown. Their values have been chosen by a tuning operation on experimental data with different dynamic behaviors. These parameters regulate smoothing action and consequently the delay of the output signal.

Processing the raw joint positions with the developed algorithm, a better signal is achieved in term of smoothness. For example, in Fig. 4, the y-coordinates of the wrist position vector for raw and processed data, related to random arm movements, are shown. The raw data exhibit jumps and discontinuities of the signal, while the processed data are free from these aspects.

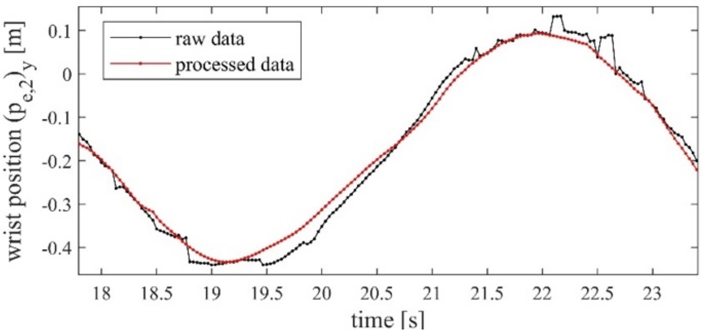


Fig. 4. y-coordinates of the wrist position vector for Kinect raw and processed data. Test with random movements

A velocity analysis is accomplished by a comparison between velocities measured with Kinect and OptiTrack. The velocity module values are calculated by numerical differentiation of the position vector. The OptiTrack signal is previously down-sampled to 30 Hz, that is the Kinect signal frequency. To ensure repeatability in the test, a constraint is provided to drive the trajectory of the hand. Fig. 5 shows the wrist velocity module measured during a motion along a straight track. The velocity measured by Kinect and processed with the proposed algorithm shows a good fitting with respect to data measured by Optitrack. The noise of the velocity raw data is correctly filtered. The delay between processed and raw Kinect data is evaluated considering relative minimum and maximum points, as displayed, in the example in Fig. 5, with the delay d. In the reported test this delay is always below 140 ms. In addition, the delay between processed Kinect data and OptiTrack data, evaluated in the same way, is always below 210 ms.

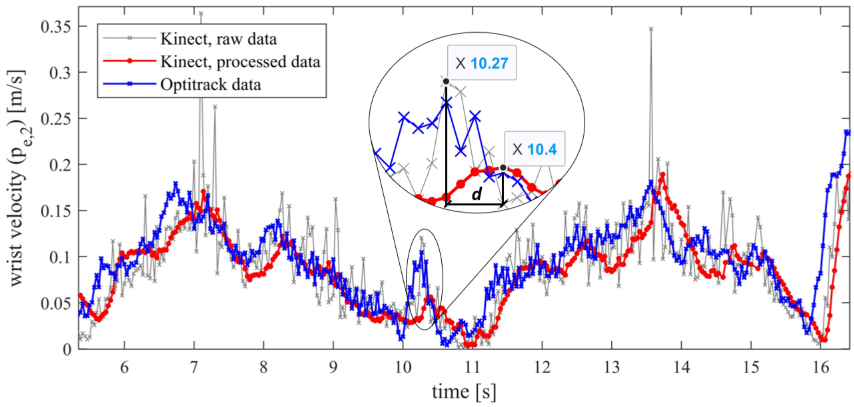


Fig. 5. Wrist absolute velocity module. A comparison between OptiTrack down-sampled data with raw and processed Kinect data. Test with linear-trajectory movements.

5 Conclusions

The presented method allows to achieve a better measurement of the human arm position and velocity using skeleton joint position raw data obtained with a Kinect sensor. The result was achieved, through a model-based algorithm, considering natural human motion constraints and using a Kalman filter on joint variables combined to the imposition of fixed distance between the joints, which is not a constraint for the Kinect native data processing algorithm. The algorithm can be easily adapted to other solutions and applications, where the human motion tracking is achieved by the extrapolation of skeletons. In order to optimize the performance, the tuning of Kalman filter parameters is a critical aspect. Future works forecast the use in coordination with wearable inertial sensors to increase the accuracy of the signal and reduce the delay in measurement. Finally, results can be further improved, through sensor fusion techniques, using more Kinect sensors at the same time.

References

1. Panagiota Tsarouchi, Sotiris Makris & George Chryssolouris (2016) Human–robot interaction review and challenges on task planning and programming, International Journal of Computer Integrated Manufacturing, 29:8, 916-931
2. M. Vasic and A. Billard, "Safety issues in human-robot interactions," 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, 2013, pp. 197-204.
3. Bernard Schmidt, Lihui Wang, Depth camera based collision avoidance via active robot control, Journal of Manufacturing Systems, Volume 33, Issue 4, 2014, 711-718,
4. L. S. Scimmi, M. Melchiorre, S. Mauro and S. P. Pastorelli, "Implementing a Vision-Based Collision Avoidance Algorithm on a UR3 Robot," 2019 23rd International Conference on Mechatronics Technology (ICMT), SALERNO, Italy, 2019.
5. Liu, H. and Wang, L. Gesture recognition for human-robot collaboration: A review, International Journal of Industrial Ergonomics, 2018, 68, 355-367.
6. Han, J.; Shao, L.; Xu, D. and Shotton J. Enhanced Computer Vision with Microsoft Kinect Sensor: A Review, IEEE Transactions on Cybernetics, 2013, 43, 1318-1334.
7. H. Ding, J. Heyn, B. Matthias and H. Staab, "Structured collaborative behavior of industrial robots in mixed human-robot environments," 2013 IEEE International Conference on Automation Science and Engineering (CASE), Madison, WI, 2013, pp. 1101-1106.
8. Microsoft Kinect SDK [Online] https://developer.microsoft.com/windows/kinect/, last accessed 2020/02/20
9. Q. Wang, G. Kurillo, F. Ofli and R. Bajcsy, "Evaluation of Pose Tracking Accuracy in the First and Second Generations of Microsoft Kinect," 2015 International Conference on Healthcare Informatics, Dallas, TX, 2015, pp. 380-389.
10. M. Melchiorre, L. S. Scimmi, S. P. Pastorelli and S. Mauro, "Collison Avoidance using Point Cloud Data Fusion from Multiple Depth Sensors: A Practical Approach," 2019 23rd International Conference on Mechatronics Technology (ICMT), SALERNO, Italy, 2019.
11. Chen, S.Y. Kalman Filter for Robot Vision: A Survey, Transaction on Industrial Electronics, 2012, 59, 4409-4420.
12. Edwards, M. and Green, R. Low-Latency Filtering of Kinect Skeleton Data for Video Game Control, IVCNZ '14 Proceedings of the 29th International Conference on Image and Vision Computing New Zealand, 2014; pp. 190-195.
13. Melchiorre, M, Scimmi, L.S., Mauro, S., Pastorelli, S. Influence of Human Limb Motion Speed in a Collaborative Hand-over Task. Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2018, Vol.2, 2018.
14. Scimmi, L.S., Melchiorre, M, Mauro, S., Pastorelli, S.: Multiple Collision Avoidance between Human Limbs and Robot Links Algorithm in Collaborative Tasks. Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2018, Vol.2, 2018.
15. Mauro, S.; Pastorelli, S.; Scimmi, L. S. (2017), Collision avoidance algorithm for collaborative robotics. In: International Journal of Automation Technology, vol. 11 n. 3, pp. 481-489. - ISSN 1881-7629, DOI 10.20965/ijat.2017.p0481
16. Brown, R. G. and P. Y. C. Hwang. Introduction to Random Signals and Applied Kalman Filtering, Fourth Edition, John Wiley & Sons, Inc., 2012
17. Y. Bar-Shalom, X. R. Li and T. Kirubarajan, Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction, Wiley, 2001
18. Forester W. Isen, DSP for MATLAB and LabVIEW I: Fundamentals of Discrete Signal Processing, Morgan & Claypool Publishers, 2008
19. OptiTrack Trio [Online] https://optitrack.com/products/v120-trio/ last accessed 2020/02/20