Scientia Iranica B (2017) 24(1), 191–201

Sharif University of Technology
Scientia Iranica
Transactions B: Mechanical Engineering
www.scientiairanica.com

Hand acceleration measurement by Kinect for rehabilitation applications

A. Mobini, S. Behzadipour* and M. Saadat Foumani

School of Mechanical Engineering, Sharif University of Technology, Tehran, Iran.

Received 14 November 2014; received in revised form 14 November 2015; accepted 12 March 2016

KEYWORDS
Kinect;
Kinematic measurement;
Acceleration measurement;
Skeleton tracking;
Filtering.

Abstract. Affordable motion sensors that are recently developed for video gaming have formed a budding line of research in the field of physical rehabilitation. These sensors have been used in many task-based applications to analyze the patients' status based on their completion of assigned tasks. However, as the accuracy of such sensors is lower than that of the clinical ones, their measured data has had very limited use in quantitative motion analysis to this date. The aim of this article is to determine Kinect's ability and accuracy in calculating higher-order kinematic parameters, such as velocity and acceleration, in hand movements. Four methods, i.e. moving average, Butterworth filter, B-spline, and Kalman filter, were proposed to calculate velocity and acceleration from Kinect's raw position data. The results were experimentally compared with two established motion capture systems, i.e. Vicon and Xsens, to analyze the strengths and weaknesses of each method. The results show that B-spline is the best method for calculating velocity and acceleration from Kinect's position data. Using this method, these parameters can be measured with an acceptable accuracy.

1. Introduction

Using economical motion-sensing game controllers like Nintendo Wii Mote, Wii Balance Board (Nintendo Inc., Redmond, Washington, USA) [1], and Microsoft Kinect (Microsoft Inc., Redmond, Washington, USA) [2] in physical and neurological rehabilitation has brought new possibilities to the field of home rehabilitation. Microsoft Kinect is in fact one of the newest technologies used for rehabilitation purposes as an input device. Early commercial rehabilitation software developed based on this sensor includes SeeMe [3], VirtualRehab [4], and JINTRONIX [5], which in general guide patients through various types of exercise in form of games and then analyze their performance based upon their task completion ratio and range of motions.

Kinect consists of an optical depth sensor, which uses a knowledge-based inference engine software that estimates the human joints' positions in 3D. The optical unit uses a speckle pattern of infrared dots to create a 3D point cloud of the object's surface [6-8]. Using this 3D map and a software using a randomized decision forest of three trees, each trained by 300,000 images, Kinect is capable of specifying body joints' positions in 30 frames per second [9].

The accuracy of Kinect's 3D map [10-12] and the ability of its software tool in estimating the position of joints have been evaluated by different researchers [13-16]. As shown in these works, Kinect's accuracy in estimating static joint positions is about 4-7 cm [16]. This accuracy is sufficient for many rehabilitation purposes as long as joint's position is the only required parameter [3-5,17].

*. *Corresponding author. Tel.: +98 21 66165542;*
Fax: +98 21 66000021
E-mail address: behzadipour@sharif.edu (S. Behzadipour)

A complete home rehabilitation system should analyze the patient's movements and assess his/her status and progress. Position derivatives such as velocity, acceleration, and jerk, also known as kinematic parameters, are of critical importance for a comprehensive analysis of human movements [18-21]. However, motion sensors usually measure only one of them, which is position in case of optical sensors and acceleration in inertial ones. Calculation of other kinematic parameters is left to the post-processing units. Since Kinect's output is relatively noisy and of low accuracy, extraction of kinematic parameters, such as velocity and acceleration, becomes cumbersome.

Clinical sensors, which are traditionally used in human motion analysis, can be divided into two groups: optical and inertial sensors. These sensors are used widely in this field by different researchers. For instance, in [20,22], an optical measurement system, Optitrack, was used to analyze hand movements of stroke patients and assess recovery process of patients. Patel et al. [19] used accelerometer data to measure hand movements of stroke patients and monitor their rehabilitation process. The main challenge in using such systems is their high cost, which makes them only feasible in research clinics.

A number of comparisons have been made on acceleration calculations by clinical optical systems such as OptiTrack (NaturalPoint, Inc., Corvallis, Oregon State, USA) and Vicon (Vicon Motion Systems, Los Angles, USA), and inertial measurement units such as Xsens (XSENS, Xsens Technologies B.V., Enschede, The Netherlands) and Konix (Konix Inc., Ithaca, New York, USA) [23-26]. In these studies, the second derivative of the position data measured by the optical motion capture system has been compared with the acceleration directly measured by the inertial sensor. As the accuracy of such motion capture systems is approximately 1 mm (about 40 times better than Kinect) and sampling frequency is more than 100 Hz (3 times higher than Kinect), a 4th order Butterworth filter provides smooth and reliable acceleration data.

The goal of this research is to extract velocity and acceleration from Kinect's position data and evaluate the accuracy of the results. In this regard, four different methods are presented for the extraction of acceleration and velocity data from Kinect's outputs. These methods are: 1) reduced moving average, 2) Butterworth filter, 3) B-spline, and 4) Kalman filter. The performance of the above methods is experimentally compared with both a well-known clinical optical system, Vicon, and a popular inertial measurement unit, Xsens, as the best available references. The results of this study are expected to help professionals in using Kinect for quantitative analysis of patient's movements.

2. Method

Extraction of higher-order kinematic parameters such as velocity and acceleration from the position data provided by Kinect is challenging due to the high level of noise in Kinect output. This noise is believed to be due to the skeleton detection algorithm built into this device [9] as well as the environmental factors affecting its optics such as infrared light in the room and the object's surface properties [10]. As a result, numerical derivation cannot be directly applied to extract velocity and acceleration.

In this paper, four methods are proposed and the results are compared in order to determine which can best estimate the higher-order kinematic parameters. Among these four methods, two are based on the idea of filtering the data before applying numerical derivation. In the other two, the natural dynamics of human movements are utilized to estimate the characteristics of the motion and, hence, no numerical derivation is applied.

In order to evaluate the four methods, an experiment was designed and carried out on a human subject. This experiment followed the literature [23,25] in using Vicon, which is a well-established optical measurement system, as the reference for acceleration and velocity data. Error analysis of each of the four proposed methods on acceleration and velocity calculation was done and the best performance was determined. Eventually, Xsens, as a well-known and relatively affordable inertial sensor for acceleration measurement, was compared with the decided best performing acceleration calculation method using Kinect.

In working with Vicon and Xsens, the standard preprocessing procedures, according to the literature, were applied for acceleration calculation [23,25,26]. As all measured data in this paper are discrete, all filters are discrete too. In the following, the proposed methods for acceleration calculation based on Kinect's output are described followed by the experimental setup.

2.1. Motion acceleration from Kinect's output

Figure 1(a) shows the Kinect's estimation of hand position versus that of Vicon as a reference. While Kinect's data follows the measurements of Vicon with a relatively good accuracy, Figure 1(b) shows that the numerical derivation of Kinect's output to find the acceleration of hand would result in significant error mostly due to noise magnification effect. In order to avoid this phenomenon, four methods are proposed and implemented, as detailed in the following sections.

Each of these methods add some delay to the calculated results. As the acceleration and velocity obtained from the Kinect's data are mostly used in offline applications (e.g., assessment of patient's state

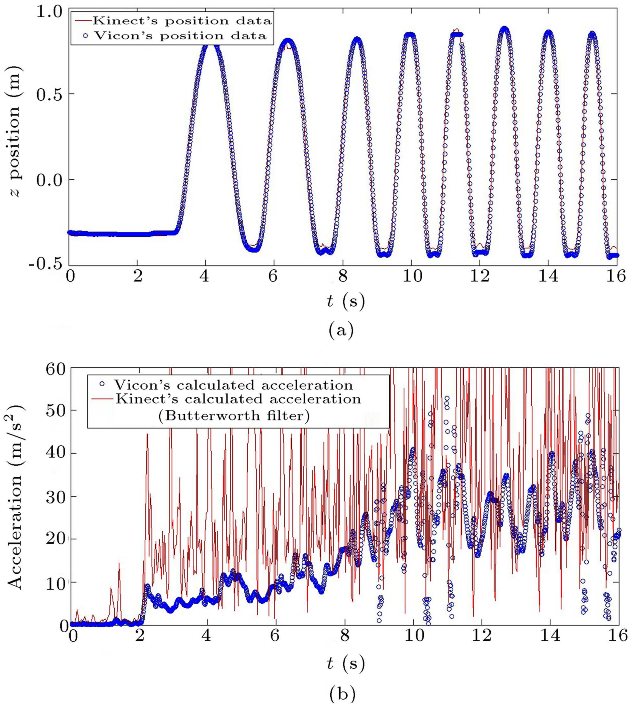

Figure 1. (a) Position and (b) acceleration measurements by Kinect and Vicon.

and recovery), the delay in measurement is not an issue. For comparing the results of Kinect with those of Vicon or Xsense, however, the two signals should be synchronized. For this purpose, the first peaks of the signals from the two sensors were matched in this work.

2.1.1. Reduced Moving Average (RMA)

The reduced moving average is a signal processing method that is quite similar to simple moving average, except in its amount of subset shifting, which is greater than one. In this method, the first element of the filtered data is obtained by taking the average of the first subset of data. Then, this subset is shifted forward by N (greater than 1) frames in order to find the next element of the filtered data. The characteristics of this filter depend on the two parameters of subset size and subset shifting (N). In this paper, a subset size of three has been used for the moving average. This subset size reduces the noise effects without any considerable lag imposed on the output. It should be noted that as the subset shifting increases, the output data become smoother and the related noise decreases, but sampling frequency decreases as well. For each system, the optimum amount of subset shifting should be determined according to the requirements of the application. In this paper, the best subset shifting was determined in Section 3 based on comparison with the reference results from Vicon.

Based on the definition of this filter, the ith element of the velocity vector is found from the following equation:

$$V_i = \frac{\frac{x_{(i+1)N-1}+x_{(i+1)N}+x_{(i+1)N+1}}{3} - \frac{x_{iN-1}+x_{iN}+x_{iN+1}}{3}}{t_i}, \tag{1}$$

where V_i stands for the velocity at the ith time step, x_i for the corresponding position value, N for the number of shifting frames, and t_i for the time interval between $x_{(i+1)N}$ and x_{iN}.

Easy application and low computational effort are among the main advantages of this method. The main drawback is its lowered sampling frequency, which may lead to losing some high-speed movements.

2.1.2. Butterworth low-pass filter

The Butterworth low-pass filter, as known from its name, is tuned to pass low-frequency signals while blocking high-frequency ones. The range of frequencies which pass through this filter is called pass band or bandwidth. It extends from $\omega = 0$ to $\omega = \omega_c$ rad/sec, where ω_c or the cutoff frequency is the highest frequency at which energy flow of the signal through the system begins to reduce. Butterworth filter is known by its transfer function as:

$$G(\omega) = \frac{G_0}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}}, \tag{2}$$

where G_0 stands for the DC gain, n for the order of the filter, and ω_c for the cutoff frequency.

This filter has been used in the literature for smoothing the position data measured by optical measurement systems such as Vicon in order to calculate the position derivatives [25,26]. For this purpose, the filter was applied to the position data before numerical derivation. The characteristics of this filter are determined using two parameters, which are the order of the filter and the cutoff frequency. In this paper, these parameters are determined through trial and error as presented in Section 3.

2.1.3. B-spline

A B-spline is a piecewise polynomial function of order k. The places where the polynomials are joined together are known as knots or breaking-points. A B-spline is a differentiable function up to the derivatives of degree $k-1$ all over its range [27]. It can be used as a function estimator for experimentally measured data [28,29]. Therefore, it can be used as an analytical approximate of the data, which can then be differentiated for finding the derivatives. As a result, it can be utilized as a filtering technique, which provides smooth and differentiable approximation of the actual data. In the application of this technique to Kinect's position data, the order of the polynomials (k) and the number of data frames between two knots (N)

may affect the filter properties. In order to obtain acceleration data, k should be greater than 3 so that the B-spline can be differentiated twice. In this article, optimum values for k and N are determined through a heuristic optimization and the calculated results are then compared with those of Vicon.

2.1.4. Kalman filter

Kalman filter is used here as a filter-observer to provide optimal estimates of a system's states, including velocity and acceleration. This filter eliminates numerical derivation and, hence, noise intensification. A Kalman filter consists of two essential components. The first one, which is the process model, is used to estimate a posteriori states (at step k) according to a priori states (at step $k-1$). This model usually employs the governing dynamic equations of the system. The second component, which is the measurement model, enables the filter to correct its estimation with respect to the measured states. The parameters that enable the filter decide on how to refine its earlier estimations are the covariance of the process and the covariance of the measurement noises. These parameters respectively represent the modeling and the measurement errors. Although the best function of a Kalman filter is expected in data fusion [30,31], there are several applications for extracting velocity and acceleration from position sensors, such as optical shaft encoders or GPS [32,33].

In this work, the state variable vector is set to include position, velocity, and acceleration:

$$X(k) = \begin{bmatrix} x & \dot{x} & \ddot{x} \end{bmatrix}^T. \tag{3}$$

The process equation represents the kinematic relation between the state vector and its derivatives. As seen in Eq. (4), an assumption has been made that acceleration is a constant variable. Since this assumption is not accurate, the modeling or the process noise ($w(k)$) is attributed to this variable.

$$\dot{X}(k) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} X(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(k). \tag{4}$$

The measurement equation determines which variables are directly measured. Since position is measured by Microsoft Kinect and Vicon, the measurement equation becomes:

$$Z(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} X(k) + v(k), \tag{5}$$

where $v(k)$ is the measurement noise. The parameters to be set are the covariance of the process error and the covariance of the measurement noise. The covariance of the measurement noise is calculated from Kinect's position error with respect to Vicon's data, as shown later in Results section. The covariance of the process error is determined from comparing Kinect's acceleration with that of Vicon. Next, the Kalman optimal observer, which minimizes the estimation error, determines the state vector. The state equation of the Kalman filter is given in Eq. (6):

$$\dot{X}_0(k) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} X_0(k) + L\left[Z(k) - \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} X_0(k)\right]. \tag{6}$$

Here, L is the Kalman gain, which has to be optimized, and X_0 is the state variables vector. By obtaining the first-order derivative of Eq. (6) and rearranging it, one will have to solve an algebraic Riccati equation to find the Kalman gain [34]. Once the Kalman gain is available, the system state equations can be solved.

2.2. Test setup and experiment

The experimental setup included three sensors: Kinect, Vicon, and Xsens. Vicon, which had six infrared cameras, and Microsoft Kinect measured position data while Xsens (Xsens MTx-28 A53 G25 sensor) measured acceleration directly.

During the test, one Xsens inertial measurement unit was strapped to the hand of the subject with one Vicon's reflective marker installed on the back of the strap. Also, one Kinect camera was placed in front of the subject in a distance of about 2 m, where it was able to record hand movements of the subject. Two reflective markers were installed on Kinect for frame registration of Kinect and Vicon. The installed markers on Kinect and the reference frame of calculations are shown in Figure 2. For frame registration of Xsens and Kinect, the Xsens sensor was placed on Kinect, parallel to Kinect's reference frame, and data was saved for about 10 seconds. Using this data, the rotation of Xsens reference frame relative to Kinect could be determined. Next, all three systems started saving data while the subject moved his hand with an increasing velocity. Hand movements were limited to vertical and diagonal directions (upper left to lower right or vice versa), as they are more common in rehabilitation applications [3-5]. Ten hand movement trials were recorded for the following analysis. In this test, the data of Vicon and Xsens was sampled at 100 Hz and Kinect recorded joint positions at 30 Hz. In order to

Figure 2. Kinect and the reflective markers.

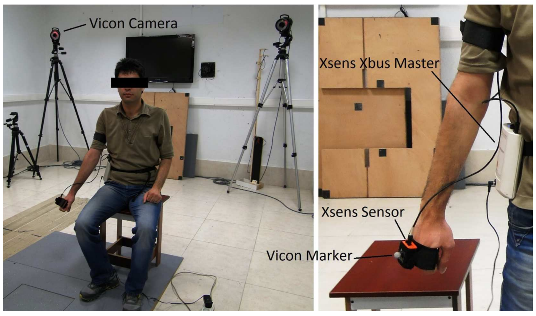

Figure 3. The test setup including Vicon, Xsense, and Kinect sensors.

estimate joint positions, Kinect uses Microsoft's Kinect for Windows SDK, which is the skeleton tracking driver provided by Microsoft [35]. Also, a program was developed in C# to record Kinect's data. A picture of the test setup is shown in Figure 3.

2.3. Acceleration measurements by Vicon and Xsens

Acceleration of hand movements was found from sensor measurements through the following steps:

1. Using the positions of the markers installed on Kinect, the transformation from Vicon's reference frame and that from Xsens to Kinect's reference frame were found (registration of Vicon's and Xsens's reference frames);
2. Vicon's measurements (markers' positions) and those of Xsens (acceleration of the IMU) were transformed to Kinect's reference frame;
3. Vicon's measurements were filtered using a fourth-order Butterworth low-pass filter with a cutoff frequency of 6 Hz. This filter had successfully been used in the previous upper limb movement studies using motion capture systems [23,25,26]. Xsens' data was also smoothed out using the same procedure [26];
4. To calculate velocity and acceleration from Vicon's measurements, numerical derivation was used following previous studies in the literature [23,25,26]. Velocity was also found from Xsens' measurements using numerical integration.

The test subject was asked to perform several hand movements while all three measurement systems were recording the movement data. These movements were devised in four categories with low and high velocity and acceleration. These categories were:

1. Vertical movement with speed < 3 m/s and acceleration < 10 m/s^2;
2. Vertical movement with speed > 3 m/s and acceleration > 10 m/s^2;
3. Diagonal movement with speed < 4 m/s and acceleration < 20 m/s^2;
4. Diagonal movement with speed > 4 m/s and acceleration > 20 m/s^2.

It should be noted that the subject could not be expected to adapt his motions to match the exact speeds and acceleration of the above classifications. To solve this problem, he was asked to start a vertical or diagonal movement from low speed and repeat it with gradually increasing speed. Later, the velocity and acceleration of all the recorded movements were calculated, and motions that belonged to any of the four categories were determined.

3. Results

Following the procedure given in the previous section, the calculated values of velocity and acceleration from Vicon and Xsens are presented here. It is noteworthy that Vicon is assumed as the reference in all procedures. Subsequently, the acceleration results from Kinect's position data are given using the proposed filters. By comparing these results with the reference data from Vicon, the final tuning of filter parameters is done. The optimized results of all filters are then compared with each other based on the reference data from Vicon.

Figure 4 shows hand velocity and acceleration in z direction calculated from the Vicon's data for movements in vertical path. As mentioned before, and shown in this figure, the velocity increased with each cycle as instructed.

3.1. Reduced Moving Average (RMA)

In Table 1, the RMS of Kinect acceleration error with respect to Vicon's results with different values of N,

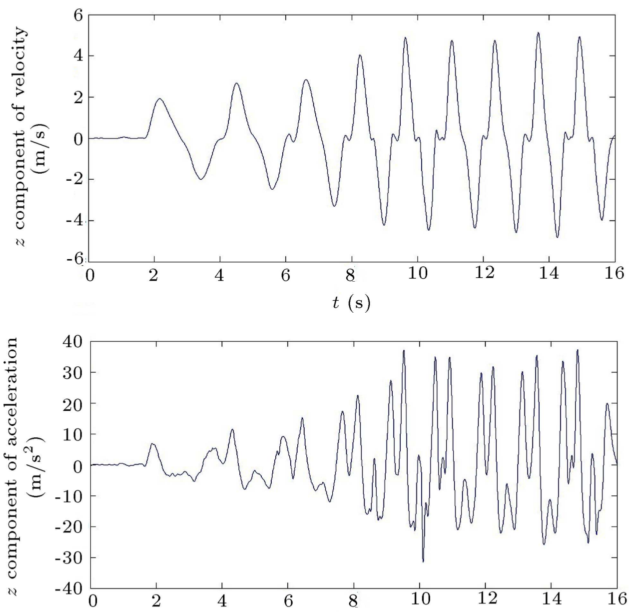

Figure 4. Vicon's calculated velocity and acceleration in z direction.

Table 1. RMS of acceleration error for reduced moving average method.

RMS of acceleration error (m/s²)					
N	**Category 1**	**Category 2**	**Category 3**	**Category 4**	**Mean**
3	1.84(18.4%)	8.6(28.6%)	4.13(20.6%)	9.15(22.8%)	22.6%
4	1.49(14.9%)	9.06(30.2%)	3.92(19.6%)	10.2(25.5%)	22.5%
5	1.61(16.1%)	8.94(29.8%)	4.08(20.4%)	12.17(30.4%)	24.1%
6	1.58(15.8%)	10.76(35.8%)	4.63(23.1%)	15.32(38.3%)	28.2%
7	1.72(17.2%)	11.68(38.9%)	5.73(28.6%)	17.05(42.6%)	31.8%
8	2.03(20.3%)	13.53(45.1%)	6.13(30.6%)	19.39(48.4%)	36.1%

Table 2. RMS of the acceleration error for Butterworth filter.

RMS of acceleration error (m/s²)						
Order	**Cutoff freq. (Hz)**	**Category 1**	**Category 2**	**Category 3**	**Category 4**	**Mean**
3	1.5	6.21 (62.1%)	13.09 (43.6%)	8.62 (43.1%)	17.12 (42.8%)	47.9%
3	3	7.04 (70.4%)	13.86 (46.2%)	9.46 (47.3%)	15.98 (39.9%)	50.9%
3	6	8.18 (81.8%)	17.20 (57.3%)	11.47 (57.3%)	29.02 (72.5%)	67.2%
4	1.5	6.34 (63.4%)	14.08 (46.9%)	8.98 (44.9%)	17.31 (43.2%)	49.6%
4	3	7.01 (70.1%)	14.05 (46.9%)	9.82 (49.1%)	16.98 (42.4%)	52.1%
4	6	7.99 (79.9%)	16.80 (56.0%)	10.98 (54.9%)	27.68 (69.2%)	64.8%
5	1.5	6.37 (63.7%)	14.99 (49.9%)	9.23 (46.1%)	17.35 (43.3%)	50.7%
5	3	6.92 (69.2%)	14.50 (48.3%)	10.05 (50.2%)	17.87 (44.8%)	53.1%
5	6	7.91 (79.1%)	16.54 (55.1%)	10.73 (53.6%)	26.86 (67.1%)	63.7%

which is the only parameter of this filter, are presented. According to this table, $N = 4$ gives the best results for movements with low speed and acceleration while $N = 3$ is the best for high values. It is seen that by increasing N, sensitivity of the filtered data to fast movements decreases. As most of the rehabilitation movements have low speed and acceleration in the following analysis, $N = 4$ is used.

Figure 5 compares Vicon's and Kinect's results for calculating acceleration in z direction using the RMA method. In lower speeds, the results are very close. However, after 6 seconds, when the movement acceleration starts increasing, the deviation between the two curves becomes apparent. This is believed to be due to the lower sampling frequency in Kinect, which results in missing the peak acceleration values.

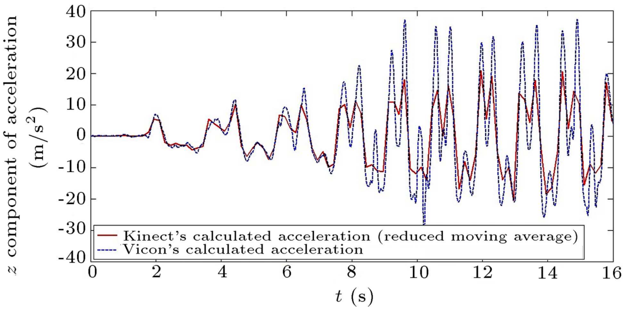

Figure 5. Vicon's and Kinect's calculated acceleration in z direction using RMA filter with $N = 4$.

3.2. Butterworth low-pass filter

In Table 2, the RMS of Kinect's acceleration error with respect to the Vicon's results is presented for different values of the filter parameters. It is seen that the filter parameters do not have any significant impact on the accuracy of the outcome. The RMS of error for all parameters in this table are relatively high, ranging from 40% to 80%.

In Figure 6, the calculated values of acceleration for Vicon and Kinect in z direction are compared. A third-order Butterworth filter with a cutoff frequency of

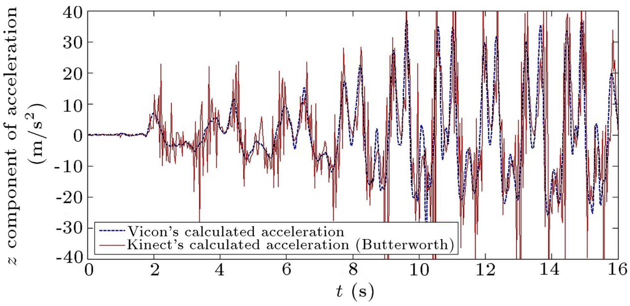

Figure 6. Vicon's and Kinect's calculated acceleration in z direction using 4th order Butterworth filter with cutoff frequency of 3 Hz.

Table 3. RMS of acceleration error for B-spline method.

		RMS of acceleration error (m/s^2)				
k	N	Category 1	Category 2	Category 3	Category 4	Mean
4	4	2.29 (22.9%)	8.52 (28.4%)	4.19 (20.9%)	11.48 (28.7%)	25.2%
4	5	1.90 (19.0%)	8.66 (28.8%)	3.78 (18.9%)	9.51 (23.7%)	22.6%
4	6	1.68 (16.8%)	9.40 (31.3%)	3.74 (18.7%)	12.06 (30.1%)	24.2%
4	7	1.46 (14.6%)	10.48 (34.9%)	5.03 (25.1%)	14.16 (35.4%)	27.5%
5	4	2.09 (20.9%)	8.15 (27.1%)	4.22 (21.1%)	8.74 (21.8%)	22.7%
5	5	1.77 (17.7%)	7.52 (25.0%)	3.02 (15.1%)	7.35 (18.3%)	19.0%
5	6	1.45 (14.5%)	8.43 (28.1%)	3.44 (17.2%)	10.48 (26.2%)	21.5%
5	7	1.42 (14.2%)	10.57 (35.2%)	3.61 (18.0%)	11.79 (29.4%)	24.2%
6	4	2.37 (23.7%)	7.98 (26.6%)	3.68 (18.4%)	7.37 (18.4%)	21.7%
6	5	1.54 (15.4%)	7.74 (25.8%)	3.19 (15.9%)	6.56 (16.4%)	18.3%
6	6	1.31 (13.1%)	8.49 (28.3%)	2.97 (14.8%)	9.55 (23.8%)	20%
6	7	1.33 (13.3%)	9.30 (31.0%)	3.14 (15.7%)	11.52 (28.8%)	22.2%
7	4	2.40 (24.0%)	8.13 (27.1%)	3.76 (18.8%)	7.99 (19.9%)	22.4%
7	5	1.59 (15.9%)	7.54 (25.1%)	3.33 (16.65%)	6.58 (16.4%)	18.5%
7	6	1.33 (13.3%)	8.01 (26.7%)	2.88 (14.4%)	8.62 (21.5%)	18.9%
7	7	1.30 (13.0%)	10.83 (36.1%)	3.44 (17.2%)	11.60 (29.0%)	23.8%
8	4	2.24 (22.4%)	7.93 (26.4%)	4.08 (20.4%)	9.40 (23.5%)	23.1%
8	5	1.67 (16.7%)	7.37 (24.5%)	3.29 (16.4%)	7.03 (17.5%)	18.7%
8	6	1.43 (14.3%)	8.28 (27.6%)	3.01 (15.0%)	7.98 (19.9%)	19.2%
8	7	1.30 (13.0%)	9.35 (31.1%)	3.75 (18.7%)	11.65 (29.1%)	22.9%

3 Hz is used here. As this figure shows, Kinect's results follow the same trend as those of Vicon. However, in addition to the high RMS of error reported in Table 2, the peak values seen in the filter output are still too high resulting in instant errors of up to 200%.

3.3. B-spline

In Table 3, the RMS of error between Kinect's results and those of Vicon are presented with different values of B-spline's parameters (i.e., the order of polynomial, k, and the number of data frames between two successive knots, N). According to this table, a B-spline of order 6 with 5 frames between two successive Knots is used in this work.

Figure 7 compares the Vicon's and Kinect's results for acceleration in z direction using the B-spline method with $K = 6$ and $N = 5$. As seen, the noise effects have been removed and Kinect's acceleration follows Vicon's results reasonably well with an average RMS error of 4.75 m/s^2. The error is particularly lower for lower speeds, which is quite expected due to the limited sampling frequency of Kinect.

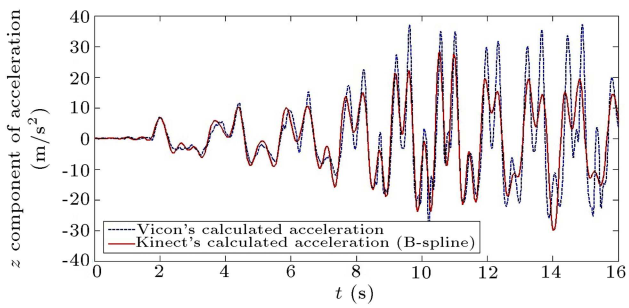

Figure 7. Vicon's and Kinect's calculated acceleration in z direction using B-spline method with $k = 6$ and $N = 5$.

3.4. Kalman filter

As discussed earlier, since the modeling assumptions are not accurate, the process noise covariance should be of a much greater order than the measurement noise covariance, which is of higher certainty. The measurement noise covariance can be easily obtained. The covariance of a signal is commonly computed by Eq. (7):

$$\mathrm{cov}(e, e^T) = \frac{1}{n}\sum_{i=1}^{n} (e_i - \bar{e})(e_i - \bar{e})^T, \tag{7}$$

where e is the noise vector, n represents its size, and $\bar{e}$ is its mean value. This equation indicates that Kinect's noise covariance is 10^{-3} m^2 taking Vicon as the reference. As for the process noise, it was seen by numerical investigation that it should be at least 10^5 times greater. As a result, the Kalman filter was applied with various values for the process noise. The final results are shown in Table 4. According to this

Table 4. Kalman filter RMS of acceleration error.

		RMS of acceleration error (m/s^2)				
Process covariance	**Measurement covariance**	**Category 1**	**Category 2**	**Category 3**	**Category 4**	**Mean**
10E2	0.001	1.61 (16.1%)	15.58 (51.9%)	7.93 (39.6%)	23.21 (58.0%)	41.4%
10E3	0.001	1.31 (13.0%)	10.86 (36.1%)	4.62 (23.1%)	16.29 (40.7%)	28.2%
10E4	0.001	1.74 (17.4%)	8.34 (28.8%)	4.65 (23.2%)	12.38 (30.9%)	25.0%
10E5	0.001	2.11 (21.1%)	9.34 (31.1%)	5.23 (26.1%)	10.24 (25.6%)	25.9%
10E6	0.001	2.32 (23.2%)	9.99 (33.3%)	5.67 (28.3%)	9.97 (24.9%)	27.4%
10E7	0.001	2.38 (23.7%)	10.34 (34.4%)	5.87 (29.3%)	10.23 (25.5%)	28.2%

Table 5. RMS of acceleration error for Kinect and Xsens.

	RMS of acceleration error (m/s^2)				
Motion	**Category 1**	**Category 2**	**Category 3**	**Category 4**	**Mean**
Xsens	0.64 (6.4%)	1.83 (7.1%)	0.65 (3.2%)	1.28 (3.2%)	4.9%
Butterworth	6.21 (62.1%)	13.09 (43.6%)	8.62 (43.1%)	17.12 (42.8%)	47.9%
B-spline	1.54 (15.4%)	7.74 (25.8%)	3.19 (15.9%)	6.56 (16.4%)	18.3%
RMA	1.49 (14.9%)	9.06 (30.2%)	3.92 (19.6%)	10.2 (25.5%)	22.5%
Kalman	1.74 (17.4%)	8.34 (28.8%)	4.65 (23.2%)	12.38 (30.9%)	25.0%

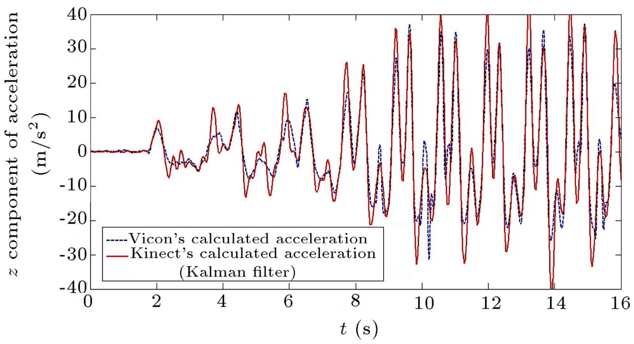

Figure 8. Vicon's and Kinect's calculated acceleration in z direction using Kalman filter.

table, the best pair is 10^4 and 10^{-3} for the process and measurement covariance, respectively.

Figure 8 compares the Vicon's and Kinect's calculated values of acceleration in z direction using the parameters found above. As shown in this figure, noise effects have been removed and the Kinect's results follow those of Vicon with a good accuracy.

4. Discussion

Thies et al. [25] have compared kinematic measurements obtained by inertial sensors and optical systems. They calculated the RMS of the difference between Xsens and Vicon measurements for forearm to be about 5%. In our experiments, this error was found to be in the range of 3 to 7% depending on the category of movement, showing a good agreement with the results of Thies et al. Taking this as an overall verification of our test procedure, the detailed discussion on the Kinect results is given in the following.

Results of acceleration calculation for different movement categories using the proposed methods are presented in Table 5. The performances of all the four methods are reported in terms of RMS error with respect to Vicon as the reference. The table shows that the B-spline method has a better performance for all ranges of movement with a mean-value RMS error of about 18%. For low-speed/acceleration movements, B-spline provides up to 31% better results compared to the other three methods. For high-speed/acceleration diagonal movements, however, the difference goes up to 47%. It is also noted that the performance of all four methods is better in low-velocity/acceleration movements (Categories 1 and 3). This indicates that the limitation of Kinect in sampling frequency is a major contributor to the error in acceleration estimation.

It is also interesting that the effect of velocity/acceleration level on the accuracy of acceleration estimation is different in vertical and diagonal motions. In vertical motion, higher velocity/acceleration results in 65%-100% higher error, while this value in diagonal motion is about 30%. This is because hand may block the Kinect's view of shoulder and elbow at some moments during the vertical movements. This leads to higher instantaneous error in tracking these joints. Another interesting observation is the poor performance of the Butterworth filter, which is the method of interest for velocity/acceleration estimation in optical systems such as Vicon [26-29]. In the case of Kinect, this method results in a mean RMS of about

Table 6. RMS of the velocity error for Kinect.

	RMS of velocity error (m/s)				
Motion	**Category 1**	**Category 2**	**Category 3**	**Category 4**	**Mean**
Xsens	0.97 (32.3%)	1.67 (33.4%)	1.49 (37.2%)	2.13 (35.5%)	34.6%
Butterworth	0.55 (18.2%)	1.37 (27.4%)	1.01 (25.2%)	1.9 (31.7%)	25.6%
B-spline	0.11 (3.7%)	0.43 (8.6%)	0.26 (6.5%)	0.48 (8%)	6.45%
RMA	0.13 (4.3%)	0.49 (9.8%)	0.40 (10%)	0.79 (13.1%)	9.27%
Kalman	0.28 (9.3%)	1.13 (22.6%)	0.72 (18.0%)	1.33 (22.1%)	18%

50%, which is the highest among the four proposed methods most probably due to the higher level of measurement noise with wider frequency spectrum. Moreover, the table indicates that the acceleration error of Kinect, even using the best filtering method (B-spline), is about four times larger than that of Xsens in all four categories of movement.

According to Figures 5-8, the performance of each method at the peak values of acceleration has a major contribution to the RMS error that is reported in Tables 5 and 6. Therefore, it is critical to understand the performance of each method at high acceleration points, which mostly corresponds to where the direction of hand movement is changed (end of trajectory). In RMA method, peak values for acceleration are always underestimations of the actual ones. In lower acceleration, they are relatively close (up to 20% lower); but as the acceleration increases, the error becomes more considerable (up to 50%). In the Butterworth, peak values are too noisy, at some points even 3 times higher than the real values. In B-spline, peak values in lower acceleration are relatively accurate (less than 20% of error). In high-acceleration movements (Categories 1 and 3), this method also underestimates the peak values by an average of 25%, which is still lower than those in the other methods. Unlike RMA and B-spline, Kalman filter tends to give an overestimation at peaks of acceleration. In both low- and high-velocity/acceleration movement categories, there is an average of 50% overestimation of acceleration. The superior performance of the B-spline method is believed to be due to its ability in capturing the motion at high-acceleration moments. In this method, a differentiable curve is fitted to the noisy data (position data) and, therefore, the noise effects are removed from the early stages of the process.

In Table 6, the RMS errors of velocity estimation using Kinect and Xsens are presented with respect to Vicon as the reference. Since calculating velocity from Xsens' acceleration data needs a numerical integration and, hence, suffers from drifting effect, the calculated velocity from Xsens shows much higher error than Kinect. Among the four proposed methods for Kinect, the B-spline filter has the best performance with a mean RMS error of 6.5% over all four categories of movements. This error is almost three times lower than acceleration estimation. Figure 9 compares the Vicon's and Kinect's results for velocity in z direction using the B-spline method. As shown in this figure, calculated velocities from the two systems are close. It is noted that the performance of all methods in estimating velocity is affected by the velocity/acceleration level of the motion almost in an identical trend to acceleration estimation as discussed above. Also, a similar difference is seen between vertical and diagonal motions (i.e., the diagonal motion is less sensitive to speed/acceleration). It can be explained similarly that in vertical motion, the Kinect's view of certain parts of shoulder may be blocked and the position error may increase accordingly.

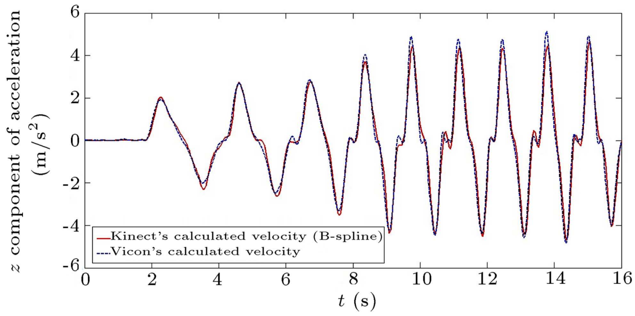

Figure 9. Vicon's and Kinect's calculated velocity in z direction using B-spline filter.

To summarize, the B-spline method proved to be the best among the four proposed methods for estimating acceleration and velocity of hand from Kinect position readings. It is believed to be mostly due to its better tracking of motions at high-speed/acceleration moments. Using this filter, acceleration can be estimated with an RMS error of 16% in lower-speed/acceleration motions and 26% in higher ones and an overall RMS error of 18%. Furthermore, movement velocities can be estimated with an overall RMS error of 6.5% in all, compared to Vicon. Also, there is always a higher chance of error in hand movements that block the shoulder joint (such as in vertical motions). It should also be noted that if computational cost is of great concern and low-speed/acceleration movements are intended, even the RMA method, which has lower computational costs, is almost as accurate as B-spline.

5. Conclusions

In this paper, Kinect's ability in the measurement of velocity and acceleration was investigated. Four methods, i.e. moving average, Butterworth filter, B-spline, and Kalman filter, were proposed to calculate velocity and acceleration from Kinect's raw position data. Kinect's calculated acceleration and velocity were compared with those of Vicon and Xsens as conventional clinical measurement systems.

Conclusive remarks are as follows:

- Using Kinect and proper filtering method, acceleration and velocity of hand movements can be measured with an acceptable accuracy;
- The results show that the B-spline filter is the best method for calculating acceleration and velocity from Kinect's data;
- In lower speeds and acceleration, Kinect can follow the movement more accurately because of its low frame rate;
- Taking Vicon as the reference, Kinect is capable of estimating acceleration and velocity of hand movement with 18.5% and 6.5% errors, respectively;
- Blocking camera view of shoulder by the hand can lower the accuracy of measurement.

Nomenclature

e	Noise vector
G	System equation
G_0	DC gain
IMU	Inertial Measurement Unit
k	Step
L	Kalman gain
n	Order of the filter
N	Number of shifting frames
RMA	Reduced Moving Average
t_i	Time interval between $x_{(i+1)N}$ and x_{iN} (s)
v	Measurement noise
V_i	Velocity at the ith time step (m/s)
w	Process noise
x_i	Position value at the ith time step (m)
X	State variable
X_0	State variable
Z	Measurement value
ω_c	Cutoff frequency (Hz)
ω	Frequency (Hz)

References

1. Nintendo Wii and balance Board [cited 2015 February]; Available from: http://www.nintendo.com/wii.
2. Microsoft Kinect [cited 2015 February]; Available from: http://www.xbox.com/en-US/kinect.
3. SeeMe System. [cited 2015 February]; Available from: http://www.virtual-reality-rehabilitation.com/products/seeme/what-is-seeme.
4. VirtualRehab System [cited 2015 February]; Available from: http://virtualrehab.info/product/#innovative-technology.
5. JINTRONIX System [cited 2015 February]; Available from: http://www.jintronix.com/.
6. Freedman, B., Shpunt, A., Machline, M. and Arieli, Y. "Depth mapping using projected patterns", WO Patent 2,008,120,217 (2008).
7. Shpunt, A. "Depth mapping using multi-beam illumination", WO Patent 2,008,087,652 (2008).
8. Spektor, E., Mor, Z., and Rais, D. "Integrated processor for 3D mapping", WO Patent 2,010,004,542 (2010).
9. Shotton, J., Sharp, T., Kipman, A., et al. "Real-time human pose recognition in parts from single depth images", *Communications of the ACM*, **56**(1), pp. 116-124 (2013).
10. Khoshelham, K. "Accuracy analysis of kinect depth data", *ISPRS Workshop Laser Scanning* (2011).
11. Dutta, T. "Evaluation of the Kinect^{TM} sensor for 3-D kinematic measurement in the workplace", *Applied Ergonomics*, **43**(4), pp. 645-649 (2012).
12. Smisek, J., Jancosek, M. and Pajdla, T. "3D with Kinect", in *Consumer Depth Cameras for Computer Vision*, Springer, pp. 3-25 (2013).
13. Pfister, A., West, A.M., Bronner, S. and Noah, J.A. "Comparative abilities of Microsoft Kinect and Vicon 3D motion capture for gait analysis", *Journal of Medical Engineering & Technology*, **38**(5), pp. 274-280 (2014).
14. Livingston, M.A., Sebastian, J., Ai, Z. and Decker, J.W. "Performance measurements for the Microsoft Kinect skeleton", *Virtual Reality Workshops (VR)*, 2012 IEEE. IEEE (2012).
15. Obdrzalek, S., Kurillo, G., Ofli, F., et al. "Accuracy and robustness of Kinect pose estimation in the context of coaching of elderly population", *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. IEEE (2012).
16. Mobini, A., Behzadipour, S. and Saadat Foumani, M. "Accuracy of Kinect's skeleton tracking for upper body rehabilitation applications", *Disability and Rehabilitation: Assistive Technology*, **9**(4), pp. 344-352 (2014).
17. Chang, Y.-J., Chen, S.-F. and Huang, J.-D. "A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities", *Research in Developmental Disabilities*, **32**(6), pp. 2566-2570 (2011).

18. Rohrer, B., Fasoli, S., Krebs, H.I., et al. "Movement smoothness changes during stroke recovery", *The Journal of Neuroscience*, **22**(18), pp. 8297-8304 (2002).
19. Patel, S., Hughes, R., Hester, T., et al. "A novel approach to monitor rehabilitation outcomes in stroke survivors using wearable technology", *Proceedings of the IEEE*, **98**(3), pp. 450-461 (2010).
20. Osu, R., Ota, K., Fujiwara, T., et al. "Quantifying the quality of hand movement in stroke patients through three-dimensional curvature", *Journal of NeuroEngineering and Rehabilitation*, **8**(1), pp. 1-14 (2011).
21. Krebs, H.I., Aisen, M.L., Volpe, B.T. and Hogan, N. "Quantization of continuous arm movements in humans with brain injury", *Proceedings of the National Academy of Sciences*, **96**(8), pp. 4645-4649 (1999).
22. Lehrer, N., Chen, Y., Duff, M., Wolf, S.L. and Rikakis, T. "Exploring the bases for a mixed reality stroke rehabilitation system, Part II: Design of interactive feedback for upper limb rehabilitation", *Journal of Neuroengineering and Rehabilitation*, **8**(1), p. 54 (2011).
23. Mayagoitia, R.E., Nene, A.V. and Veltink, P.H. "Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems", *Journal of Biomechanics*, **35**(4), pp. 537-542 (2002).
24. Skogstad, S.A., Nymoen, K. and Hovin, M. "Comparing inertial and optical mocap technologies for synthesis control", *Proc. of Int. Sound and Music Computing Conference* (2011).
25. Thies, S., Tresadern, P., Kenney, L., et al. "Comparison of linear accelerations from three measurement systems during "reach & grasp"", *Medical Engineering & Physics*, **29**(9), pp. 967-972 (2007).
26. Tresadern, P.A., Rigby, J., Goulermas, J.Y., et al. "Simulating acceleration from stereophotogrammetry for medical device design", *Journal of Biomechanical Engineering*, **131**(6), p. 061002 (2009).
27. De Boor, C., *A Practical Guide to Splines*, **27**, New York, Springer-Verlag (1978).
28. Gans, P. and Gill, J.B. "Smoothing and differentiation of spectroscopic curves using spline functions", *Appl. Spectrosc.*, **38**(3), pp. 370-376. (1984).
29. Vicsek, M., Neal, S.L. and Warner, I.M. "Time-domain filtering of two-dimensional fluorescence data", *Appl. Spectrosc.*, **40**(4), pp. 542-548 (1986).
30. Caron, F., Duflos, E., Pomorski, D. and Vanheeghe, P. "GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects", *Information Fusion*, **7**(2), pp. 221-230 (2006).
31. Leung, K.T., Whidborne, J.F., Purdy, D. and Barber, P. "Road vehicle state estimation using low-cost GPS/INS", *Mechanical Systems and Signal Processing*, **25**(6), pp. 1988-2004 (2011).
32. Shaowei, W. and Shannming, W. "Velocity and acceleration computations by single-dimensional Kalman filter with adaptive noise variance", *Przegląd Elektrotechniczny (Electrical Review)* (2012).
33. Zimmerman, Y., Oshman, Y. and Brandes, A. "Improving the accuracy of analog encoders via Kalman filtering", *Control Engineering Practice*, **14**(4), pp. 337-350 (2006).
34. Zarchan, P. and Musoff, H., *Fundamentals of Kalman Filtering: A Practical Approach*, **208**, Aiaa (2005).
35. Microsoft Kinect For Windows, Accessed July 2014 [cited 2014 June]; Available from: http://www.microsoft.com/en-us/kinectforwindows/.

Biographies

Amir Mobini obtained his BSc degree in Mechanical Engineering from Amirkabir University of Technology in 2004, Tehran, Iran. He received his MSc and PhD degrees in Mechanical Engineering from Sharif University of Technology in 2007 and 2015, respectively. His research interests include designing mechanical systems, robotics, and smart rehabilitation devices.

Saeed Behzadipour obtained his BSc and MSc degrees in Mechanical Engineering from Sharif University of Technology Tehran, Iran, in 1998 and 2000, respectively. He received his PhD in Mechanical Engineering from University of Alberta in 2005. From 2005 to 2011, he was with University of Alberta as Assistant Professor. He is now an Associate Professor with Sharif University of Technology. His research interests include medical robotics and smart rehabilitation devices.

Mahmoud Saadat Foumani received his PhD degree in Mechanical Engineering from Sharif University of Technology, Tehran, Iran, in 2002. He was a Faculty Member at Semnan University from 2002 to 2006 and is now a Faculty Member in Mechanical Engineering Department at Sharif University of Technology. He teaches courses in the 'Applied Design Group' at undergraduate and graduate levels. His teaching focuses on mechanical engineering design, vehicle dynamics and chassis design, and advanced mathematics.