Article

Estimation of Displacement for Internet of Things Applications with Kalman Filter

Renato Ferrero *,† , **Filippo Gandino** † **and and Masoud Hemmatpour** †

Dipartimento di Automatica e Informatica, Politecnico di Torino, 10129 Torino TO, Italy

* Correspondence: renato.ferrero@polito.it; Tel.: +39-0110-907-177

† These authors contributed equally to this work.

Abstract: In the vision of the Internet of Things, an object embedded in the physical world is recognizable and becomes smart by communicating data about itself and by accessing aggregate information from other devices. One of the most useful types of information for interactions among objects regards their movement. Mobile devices can infer their position by exploiting an embedded accelerometer. However, the double integration of the acceleration may not guarantee a reliable estimation of the displacement of the device (i.e., the difference in the new location). In fact, noise and measurement errors dramatically affect the assessment. This paper investigates the benefits and drawbacks of the use of the Kalman filter as a correction technique to achieve more precise estimation of displacement. The approach is evaluated with two accelerometers embedded in commercial devices: A smartphone and a sensor platform. The results show that the technique based on the Kalman filter dramatically reduces the percentage error, in comparison to the assessment made by double integration of the acceleration data; in particular, the precision is improved by up to 72%. At the same time, the computational overhead due to the Kalman filter can be assumed to be negligible in almost all application scenarios.

Keywords: accelerometer; Kalman filter; position tracking; displacement

1. Introduction

The aim of the Internet of Things is to trace a map of the real world, giving an electronic identity to physical objects and places [1]. This enables the inter-connectivity of devices: The objects gain new and powerful capabilities by sharing data with each other. In this scenario, the position of the connected objects is often an essential piece of information for effective representation of the real world. Without a central co-ordinator, every device should autonomously trace its own movements. A low-cost and accessible method relies on the use of an embedded accelerometer.

An accelerometer measures the proper acceleration of an object [2]. The proper acceleration is relative to an inertial stationary observer; for example, the gravity acting upon the inertial observer is not considered in the proper acceleration. For this reason, an accelerometer in free-fall will measure zero, whereas an accelerometer at rest will measure an acceleration of 1 g due to the Earth's gravity. Internally, the accelerometer can be considered as a damped mass on a spring. The deformation of the spring gives a measurement of the acceleration: In a resistive or capacitive accelerometer, the deformation is converted into a change of the impedance, whereas, in a piezo-electric accelerometer, it is expressed as an electrical signal.

The motion and displacement (i.e., change of position) of an object can be analyzed by measuring its acceleration. In fact, the acceleration can reveal the intensity and the frequency of the movement, as it is proportional to the external force applied to the object [3]. Basically, integration (with respect to time) of the values provided by the accelerometer gives information about the speed and displacement

of the object. In addition, information about inclination can be retrieved, such as the orientation and tilt of an object rotating on a reference plane [4]. Furthermore, wearable devices commonly exploit accelerometers to evaluate human actions in vehicle control [5], clinical and lab environments [6], and in a free-living context [7]. The first example of using an accelerometer to analyze human motion was presented in the 1950s [8]. This trend was accelerated by technological improvements in the 1970s [9]; thanks to micro-electro-mechanical systems (MEMS), sensor performance was notably increased while both cost overhead and power requirements were significantly reduced. In 1979, the first MEMS accelerometers were batch-fabricated [10] and commonly integrated into wearable objects for the analysis of human activity and movements, both for research and commercial purposes [7,11]. The size and cost of accelerometers have continued to decrease, due to further advances in integrated MEMS, without compromising quality and reliability standards [12].

This paper investigates movement tracking by means of a single accelerometer. In particular, the goal is computation of the displacement based on the data collected by a three-axis linear accelerometer. Typically, the displacement is estimated by a double integration of the acceleration data. First, the speed is computed by integrating the acceleration, following which the displacement is retrieved by integrating the speed. However, measurements are affected by errors and noise, which dramatically reduces the precision of the results. This paper analyzes a countermeasure for this issue, based on the Kalman filter [13,14]. The Kalman filter is commonly exploited in dynamic systems, in order to make estimations based on uncertain information. This paper extends a preliminary analysis [15] by comparing the performance of two different accelerometers and by evaluating the overhead introduced by the Kalman filter. The main goal of this paper is to evaluate the benefits of the application of the Kalman filter. Different experiments were made, studying movements with one, two, and three degrees of freedom. The collected statistics reveal that the technique based on the Kalman filter produces more accurate results, with a reduction of the estimation error ranging between 18–72%. Moreover, although the Kalman filter requires additional computation time with respect to the integration of the acceleration values, the overhead can be assumed to be negligible.

The remainder of the paper is organized as follows. The related works for estimating a displacement according to the values produced by an accelerometer are reviewed in Section 2. The Kalman filter and the approach to exploit it for estimating the displacement are described in Section 3. Section 4 describes the methodology followed to evaluate the accuracy of the developed approach. The experimental analysis is provided in Section 5. Section 6 investigates the overhead due to the proposed approach. Finally, some conclusions are drawn in Section 7.

2. Related Work

A common approach for retrieving position information is by integration of the signal acquired from an accelerometer [16]. However, this method is reliable only without strict requirements about the result precision, as the input data usually contain noise and measurement errors that need to be properly managed. The correction techniques proposed in the literature are targeted to the specific application scenario.

Human movement can be tracked by means of inertial and magnetic sensors on body parts, such as limbs and feet [17]. Cyclic phases can be frequently detected in various kinds of human actions, such as walking, running, and side-steping. For instance, walking is composed of an alternation of stance and swing phases. In some periodic moments, the instantaneous value of the speed may be zero (e.g., between the stance and the swing phases, when both feet touch the ground). Recognizing these events may increase the accuracy of the results. In these cases, if the result of integration of the acceleration is not null, as expected, the error of the drift in the measurements can help to correct the accelerometer results, by providing consistent speed data. Consequently, better data on human movement can be provided; for example, the average error measured by tests on indoor walking has been recorded to be 5.5% [17].

In automotive engineering, displacement is commonly estimated by means of inertial-based measurement techniques [18]. In fact, due to their small dimensions, accelerometers can be used in a vehicle. Furthermore, information about the structural integrity of the vehicle can be obtained by studying the accelerometer data: For an elastic structure, the stress responsible for structural damage is proportional to the displacement. Finally, vibrations can be properly detected by the wide frequency and dynamic range of acceleration. Three digital filtering approaches have been designed to reduce the integration errors in low-frequency random acceleration data related to on-road vibration patterns and vehicle movements. Filtering approaches exploit the response of the finite and the infinite impulses with the inverse of the fast Fourier transform [18]. With their adoption, the standard error in the displacement estimation has been shown to range from 5.7% at 20 Hz to 21.5% at 150 Hz [18].

An off-line analysis of the displacement which uses the acceleration measurements has been provided for damage detection in structural health monitoring [19]. A variation of the Kalman filter has been applied to limit the increase in noise due to the integration of the acceleration values. In particular, the condition at a specific point is estimated off-line by knowing the entire sequence of measurements. Kalman filter smoothing is introduced by analyzing a Kalman filter applied to all the samples, backward and forward. A multi-rate Kalman filter should be applied, due to the different sampling frequencies used in civil engineering applications. The displacement error is affected by the acceleration values (e.g., considering a swept-sine signal, the root mean square error is equal to 2.5% [19]).

An alternative approach to computing the displacement is realized through conversion of the acceleration values to the frequency domain [20,21]. Given a discrete series of acceleration samples $\{a_1, a_2, \ldots, a_k, \ldots, a_N\}$, the discrete Fourier transform is applied for the conversion. The discrete Fourier transform A_k of an acceleration sample a_k is:

$$A_k = \sum_{k=0}^{N-1} a_k e^{-i\frac{2\pi}{N}kr}, \tag{1}$$

with i corresponding to the imaginary unit. The discrete Fourier transform D_k of the kth displacement is immediate, due to the property of the Fourier transform of the integrals:

$$D_k = -\frac{1}{(2\pi k)^2} A_k . \tag{2}$$

It is possible to take the inverse Fourier transform of the coefficients of (2) in order to obtain the discrete series of the displacement:

$$d_r = \sum_{k=0}^{N-1} D_k e^{i\frac{2\pi}{N}kr} . \tag{3}$$

The scale factor of the error involved in (3) is $-1/(2\pi k)^2$. Consequently, when the frequency is doubled, the displacement is reduced by 6 dB. Thus, the components of the dominant signal may be lower than those at low-frequency noise during conversion. Low-frequency components should be zero padded, in order to be attenuated, with the exception of the significant sinusoidal components within the Fourier transform of the acceleration signal [20]. A better technique has been proposed, based on the significant frequency components of the Fourier transform of acceleration values before applying inverse-Fourier transform [21]. A curve-fitting correction on the local maximum points of the discrete Fourier transform can be applied to execute this operation. Curve-fitting corresponds to the finding which coefficients of a function better match a series of points.

Not only the error in the integration of the acceleration values, but also the drift and other noise effects, as well as double integration of thermal noise, affect the estimation of the displacement [22,23]. This effect can be checked with a fixed accelerometer: A long integration time increases the root mean square value, which is even affected by the sampling frequency. Initial calibration of an accelerometer

can exploit the theoretical model of the described error, taking into account both analytical [22] and numerical [23] integration.

3. Theoretical Approach

The speed of an object at time T can be obtained by integrating the acceleration over time, if its initial speed is zero. The speed can be computed as a summation of samples at discrete instants t_s:

$$v(T) = \int_0^T a(t)dt = \lim_{n \to \infty} \frac{T}{n} \sum_{s=1}^n a(t_s) \,. \tag{4}$$

The end of (4) reveals that the speed corresponds to the sum of the area below the acceleration curve in the Cartesian plane. Algorithm 1 follows this approach to estimate the speed of an object. The input is the acceleration along an axis, obtained by querying the accelerometer with the acceleration.getx() pseudo-command, and the output is the value of the speed along the same axis. Algorithm 1 is repeated as many times as the degrees of freedom of the movement; for example, the speed of an object moving in three-dimensional space is obtained by three iterations of the algorithm. The variables $speed\_initial$, $speed\_new$, and $time$ all start at zero. Spurious data produced by the accelerometer are discarded after comparison with admissible values. The threshold for determining an acceptable value is different for all the axes and is related to the values of acceleration; for example, the threshold applied to z-axis is greater than that applied to the other axes, as it also includes the gravitational force.

Algorithm 1 Computation of speed along the x-axis.

1: $speed\_initial_x = speed\_new_x$

2: $time = time + \Delta t$

3: $acc_x =$ acceleration.getx()

4: **if** $acc_x < threshold\_low$ **then**

5:    $acc_x = threshold\_low$

6: **end if**

7: **if** $acc_x > threshold\_high$ **then**

8:    $acc_x = threshold\_high$

9: **end if**

10: $speed\_new_x = speed\_initial_x + acc_x \cdot \Delta t$

By definition, speed is the distance covered per unit of time. Algorithm 2 approximates the integral of the speed by a summation:

$$d(T) = \int_0^T v(t)dt = \lim_{n \to \infty} \frac{T}{n} \sum_{s=1}^n v(t_s) \,. \tag{5}$$

A significant difference between the value computed in (5) and the real displacement can occur, due to noise and measurement errors. This gap can be limited by means of a Kalman filter [13], which statistically minimizes the estimation error. The acquisition of a new measurement does not require that any previous data are processed again. During the data-processing flow of the algorithm, input data (i.e., discrete-time measurement samples) are first acquired, then they are processed, and finally new output data are produced.

Algorithm 2 Computation of displacement along the *x*-axis.

1: $x\_initial = x\_new$

2: **call** Algorithm 1

3: $d_x = (speed\_initial_x + speed\_new_x)/2 \cdot \Delta t$

4: $x\_new = x\_initial + d_x$

The Kalman filter evaluates the system model, row data, and data uncertainty [14]. The system model can be described by a linear stochastic difference equation

$$x_k = Ax_{k-1} + Bu_k + w_k, \tag{6}$$

where $x_k \in \mathbb{R}^n$ represents the state of the system at time k, A is an $n \times n$ matrix that indicates how the system evolves to the next step without any process noise, $u_k \in \mathbb{R}^l$ is the optional control input, B is an $n \times l$ matrix that represents the control-input model applied to u_k, and w_k is the process noise. At the instant k, a measurement $z_k \in \mathbb{R}^m$ of the true state x_k is made, according to

$$z_k = Hx_k + v_k, \tag{7}$$

where H is an $m \times n$ matrix that links the true state x_k to the observed state z_k; v_k is the measurement noise, which is considered to be white and Gaussian (i.e., not correlated in time and with probability density equal to the normal distribution); and Q and R are the matrices of covariance of the process and measurement noise. In a covariance matrix, the element (i, j) corresponds to the covariance between the ith and jth elements of the source vector. In other words, a covariance matrix reports the covariance between any two elements of the vectors.

The a priori state estimate at time k is defined as the estimation obtained by relying only the process before time k. It is indicated as $\hat{x}_{k|k-1} \in \mathbb{R}^n$. On the contrary, the a posteriori state estimate at time k exploits also the knowledge of the measurement z_k to estimate the state. It is indicated as $\hat{x}_{k|k} \in \mathbb{R}^n$. The a priori and a posteriori estimate errors are

$$\begin{aligned} e_{k|k-1} &= x_k - \hat{x}_{k|k-1} \\ e_{k|k} &= x_k - \hat{x}_{k|k}, \end{aligned} \tag{8}$$

and the a priori and a posteriori estimate error covariance matrices are:

$$\begin{aligned} P_{k|k-1} &= E[e_{k|k-1} \cdot e_{k|k-1}^T] \\ P_{k|k} &= E[e_{k|k} \cdot e_{k|k}^T] \, . \end{aligned} \tag{9}$$

The Kalman filter estimates the process with a feedback control. In the predict phase, the process at time k is estimated without knowledge of z_k. Then, in the update phase the estimation is refined by including knowledge of z_k. The a priori estimations of the state and the error covariance matrix are computed in the predict phase, as follows

$$\begin{aligned} \hat{x}_{k|k-1} &= A\hat{x}_{k-1|k-1} + Bu_k \\ P_{k|k-1} &= AP_{k-1|k-1}A^T + Q \, . \end{aligned} \tag{10}$$

The feedback control is implemented in the update phase by the sum of the new measurement with the a priori estimate, in order to obtain a better a posteriori estimate:

$$\begin{aligned} \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}) \\ P_{k|k} &= (I - K_kH)P_{k|k-1}, \end{aligned} \tag{11}$$

where K, which is the Kalman gain, is a $n \times m$ matrix that minimizes the a posteriori error covariance:

$$K_k = \frac{P_{k|k-1}H^T}{HP_{k|k-1}H^T + R} \; .$$

(12)

Algorithm 3 shows an implementation of the Kalman filter for computation of the speed on the x-axis. Every iteration of the algorithm computes the a priori and a posteriori estimates of the speed at the current time. In particular, the a priori estimate of the speed is based on the a posteriori estimate from the previous iteration. In more detail, the a priori estimate is computed in the predict phase by multiplying the previous a posteriori estimate by a constant A, because there are no control inputs. In a similar way, the a priori estimate of the error covariance uses the a posteriori estimate of the error covariance from the previous iteration of the algorithm. The a priori estimate of the error covariance is applied in the update phase for the computation of the Kalman gain. The speed value used in this phase is the last value computed by Algorithm 1. Finally, the a posteriori estimate of the speed and error covariance are updated, as they are needed for the subsequent iteration.

Algorithm 3 Use of the Kalman filter to compute speed along the x-axis.

1: $priori\_speed_x = A \cdot posteriori\_speed_x$

2: $priori\_P_x = A \cdot posteriori\_P_x \cdot A^T + Q$

3: $z\_speed_x = speed\_new_x$

4: $K = priori\_P_x \cdot H^T / (H \cdot priori\_P_x \cdot H^T + R)$

5: $posteriori\_speed_x = priori\_speed_x + K \cdot (z\_speed_x - H \cdot priori\_speed_x)$

6: $posteriori\_P_x = (1 - K \cdot H) \cdot priori\_P_x$

Algorithm 4, which computes the displacement on the x-axis, is similar to Algorithm 3. Both algorithms are executed three times, in order to compute speed and displacement along each of the three axes.

Algorithm 4 Use of the Kalman filter to compute displacement along the x-axis.

1: $priori\_d_x = A \cdot posteriori\_d_x$

2: $priori\_P_x = A \cdot posteriori\_P_x \cdot A^T + Q$

3: $z\_x = x\_new$

4: $K = priori\_P_x \cdot H^T / (H \cdot priori\_P_x \cdot H^T + R)$

5: $posteriori\_d_x = priori\_d_x + K \cdot (z\_x - H \cdot priori\_d_x)$

6: $posteriori\_P_x = (1 - K \cdot H) \cdot priori\_P_x$

---

4. Evaluation Scenario

The method developed in Section 3 for the computation of the displacement was evaluated with two accelerometers produced by ST Microelectronics: LIS344ALH [24] and LIS331DLH [25]. The IC interface of the accelerometers acquires information from the sensing element and provides an output signal. Suspended silicon structures inside the sensing element are able to move along the direction of the sensed acceleration. Therefore, when an acceleration is applied, the proof mass changes, which generates an imbalance in the capacitive circuit. A charge integration can measure the imbalance by applying a voltage pulse to the sense capacitor.

The LIS344ALH is the accelerometer available in the Waspmote, an Arduino-based wireless sensor platform commercialized by Libelium [26] and shown in Figure 1. It is a ultra compact, low-power, analog output three-axis linear accelerometer. The user can dynamically select the acceleration full scale, choosing between ± 2 g and ± 6 g. The maximum bandwidth of the measured acceleration is 1.8 kHz. The IC interface processes the signal acquired from the sensing element by means of

a low-noise capacitive amplifier and three different sample and hold circuits. The analog signal provided by the IC interface is an offset voltage, which is ratiometric to the voltage supply V_{dd}; the offset increases or decreases linearly, according to the change of the voltage supply. Furthermore, the sensitivity, which indicates the gain of the sensor, is ratiometric to the voltage supply. The sensitivity is obtained by performing one measure and by repeating the same measurement when 1 g is applied to the sensor. This can be easily obtained by pointing the sensor towards the center of the Earth, and then repeating the measurement with the sensor rotated by 180 degrees, in order to add the gravitational force. The mean of the two output values provided by the sensor gives its sensitivity. With $V_{dd} = 3.3$ V, the sensitivity of the LIS344ALH ranges between $V_{dd}/5$ and $V_{dd}/15$ V/g and its change, according to the temperature, is ±0.01 %/$^\circ$C.

Figure 1. The Waspmote with the embedded accelerometer used in the experiments.

The LIS331DLH is the accelerometer embedded in the iPhone 4S smartphone. The LIS331DLH is a nano, low-power, digital output three-axis linear accelerometer, in which the three full scales of ±2 g, ±4 g, and ±8 g are available. The output data rate is between 0.5 Hz to 1 kHz. The technology of the sensing element is the same as the LIS344ALH accelerometer. The IC interface exploits a low-noise capacitive amplifier to generate an analog voltage, following which an analog-to-digital converter provides a discrete-time signal. The output data can be easily accessed by means of a I^2C/SPI serial interface. The typical voltage supply is $V_{dd} = 2.5$ V. The analog-to-digital converter uses a 12 bit representation; the sensitivity ranges between 0.9–4.3 m g/digit and its change, according to the temperature, is ±0.01%/$^\circ$C.

For the experiment with the LIS344ALH accelerometer, a program developed in C for Arduino was compiled and uploaded to the Waspmote. The program acquires data from the accelerometer and computes the displacement, according to two algorithms: The basic double integration of the input data and the application of the Kalman filter to the input data. A similar application was developed with the Swift programming language for iOS and uploaded onto the iPhone 4S, in order to repeat the experiment with the LIS331DLH accelerometer. Motion data were received from the device hardware and processed by means of the Core Motion framework.

In every experiment, the distance between an initial and a final point was first evaluated with a measuring tape. Then, both devices were moved along a straight path between the two points. The initial speed of each of the accelerometers was zero, and their speed was randomly changed during the movement. Three kinds of paths were considered, with one, two, and three degrees of freedom. In the first case, the *x*-axis of the accelerometer was aligned with the straight line followed by the device. In the second case, the device moved over a desk, which coincided with the xy-plane of the accelerometer. In the third case, the device freely moved in the space and the accelerometer recorded data along all three axes.

Three reference distances were considered in the experiments: 50, 70, and 100 cm. The devices covered every distance $N = 3$ times. Each estimation e_i of the displacement d was recorded and the following statistics were computed:

- The mean e gives a measure of central tendency:

$$\bar{e} = \frac{1}{N} \sum_{i=1}^{N} e_i \ . \tag{13}$$

- The range R indicates how widely spread the estimations are:

$$R = e_{max} - e_{min} \ . \tag{14}$$

- The uncertainty in a measurement Δe suggests the precision of the estimations. If a lot of data are available, the uncertainty coincides with the standard deviation:

$$\Delta e = \sigma = \sqrt{\frac{\sum_{i=1}^{N}(e_i - \bar{e})^2}{N}} \ . \tag{15}$$

According to (15), the majority of the data lies in the range $\bar{e} \pm \Delta e$. With a small number of data, the uncertainty can be computed as:

$$\Delta e = \frac{R}{2} = \frac{e_{max} - e_{min}}{2} \ . \tag{16}$$

According to (16), the displacement ranges in the interval between e_{max} and e_{min}. As each experiment was repeated only three times, the uncertainty in a measurement was computed as in (16).

- The uncertainty in the mean $\Delta \bar{e}$ indicates how far the displacement is from the average value of the estimations:

$$\Delta \bar{e} = \frac{\Delta e}{\sqrt{N}} \ . \tag{17}$$

Substituting (16) into (17), if few data are available, the uncertainty in the mean can be computed as:

$$\Delta \bar{e} = \frac{R}{2\sqrt{N}} \ . \tag{18}$$

- The mean absolute percentage error $MAPE$ expresses the discrepancy between the displacement and the estimations:

$$MAPE = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{d - e_i}{d} \right| \ . \tag{19}$$

All statistics were computed both for the estimation based on integration of the acceleration data and for the estimation based on the Kalman filter. Finally, in order to quantify the benefits introduced by the Kalman filter, the percentage decrease of the MAPE was computed, as follows:

$$decrease = 100 \cdot \frac{MAPE_{acceleration\ integral} - MAPE_{Kalman\ filter}}{MAPE_{acceleration\ integral}}. \tag{20}$$

5. Experimental Results

Tables 1 and 2 present the statistics of the experiment with one degree of freedom, when the acceleration data were recorded with the Waspmote and the smartphone, respectively. The estimation was usually higher than the real displacement: This can be noted by observing that the mean value of the estimation was always higher than the distance covered by the device. However, some instances of

the experiment provided an output value lower than the real displacement. This can be easily detected by checking the range of the estimations. For example, the range of the values computed with a double integration of the data obtained from the Waspmote was always higher than the displacement; this means that the maximum estimation was higher than the displacement, but the minimum one was lower. On the contrary, a narrow range indicates that the single estimations were closer to the mean value. For example, this happened in the trials with 50 and 70 cm, both with the Waspmote and the smartphone. In these cases, all estimations were higher than the path covered, as their average value was higher, but not by too much, since their range was tight. A correction can be easily applied to reach a better estimation; for example, by applying a reductive factor to the mean value or by giving more weight to the minimum estimation. By comparing the mean absolute percentage errors, it can be noted that the use of the Kalman filter decreased the percentage error in the estimation at least by 50%. However, only the accelerometer of the Waspmote allowed for precise results, with an acceptable error.

Table 1. Displacement along the *x*-axis measured with the Waspmote.

	Acceleration Integral			With Kalman Filter		
Real distance (cm)	50	70	100	50	70	100
Mean (cm)	74.3	127.0	178.6	59.6	84.0	118.0
Range (cm)	76	82	141	23	53	94
Uncertainty in measure (cm)	38.0	41.0	70.5	11.5	26.5	47.0
Uncertainty in mean (cm)	21.9	23.6	40.7	6.6	15.3	27.1
Percentage error (%)	68.7	81.9	78.7	19.3	27.6	38.6
Percentage decrease (%)	-	-	-	71.8	66.3	50.8

Table 2. Displacement along the *x*-axis measured with the smartphone.

	Acceleration Integral			With Kalman Filter		
Real distance (cm)	50	70	100	50	70	100
mean (cm)	120.0	119.3	177.3	71.6	78.6	107.6
Range (cm)	29	58	199	43	38	138
Uncertainty in measure (cm)	14.5	29.0	99.5	21.5	19.0	69.0
Uncertainty in mean (cm)	8.3	16.7	57.5	12.4	10.9	39.8
Percentage error (%)	140.0	70.5	95.3	51.3	27.6	47.7
Percentage decrease (%)	-	-	-	63.3	60.8	50.0

Data concerning the movement with two degrees of freedom is listed in Tables 3 and 4 for the Waspmote and the smartphone, respectively. Similarly to the previous set of trials, the average estimation was always higher than the real distance covered by the device. It can be noted that the average value obtained with a double integration of the acceleration data could not predict the displacement. In fact, even if the average value was decreased by the uncertainty in the mean, the result was still higher than the real displacement. By applying the Kalman filter, the uncertainty in the mean was lower; which means that the estimations were closer to each other. At the same time, the estimations based on the data collected with the accelerometer embedded in the Waspmote were closer to the real distance, so a satisfactory match can be obtained by correcting the average value with the uncertainty in the mean. The only exception, according to Table 3, occurs when the path covered was 70 cm. In contrast, relying on the mean and on its uncertainty was not enough if the Kalman filter was used with the data obtained from the smartphone. However, the use of the Kalman filter is always suggested, as it reduced the percentage error of the estimation in every case.

Table 3. Displacement along the xy-plane measured with the Waspmote.

	Acceleration Integral			With Kalman Filter		
Real distance (cm)	50	70	100	50	70	100
Mean (cm)	79.6	127.6	120.6	56.3	84.6	103.3
Range (cm)	56	44	43	38	26	17
Uncertainty in measure (cm)	28.0	22.0	21.5	19.0	13.0	8.5
Uncertainty in mean (cm)	16.1	12.7	12.4	10.9	7.5	4.9
Percentage error (%)	59.3	82.4	20.7	26.0	21.9	6.0
Percentage decrease (%)	-	-	-	56.2	73.4	71.0

Table 4. Displacement along the xy-plane measured with the smartphone.

	Acceleration Integral			With Kalman Filter		
Real distance (cm)	50	70	100	50	70	100
Mean (cm)	90.0	83.6	176.6	79.3	81.0	144.3
Range (cm)	22	36	69	13	22	63
Uncertainty in measure (cm)	11.0	18.0	34.5	6.5	11.0	31.5
Uncertainty in mean (cm)	6.3	10.4	19.9	3.7	6.3	18.2
Percentage error (%)	80.0	22.4	76.7	58.7	15.7	44.3
Percentage decrease (%)	-	-	-	26.7	29.8	42.2

The data shown in Tables 5 and 6, about the movement with three degrees of freedom with the Waspmote and smartphone, respectively, confirm the general trend of the previous experiments. Notably, the estimations were always higher than the measured displacement. However, the Kalman filter reduced the gap between the average value of the estimation and the displacement, both with the Waspmote and the smartphone. Similarly, the percentage error of the single estimation always decreased by applying the Kalman filter to the acceleration data. The accelerometer embedded in the smartphone provided results almost precise as those obtained from the accelerometer of the Waspmote, independently of the use of the Kalman filter.

Table 5. Displacement in the space measured with the Waspmote.

	Acceleration Integral			With Kalman Filter		
Real distance (cm)	50	70	100	50	70	100
Mean (cm)	89.3	91.3	160.6	64.3	84.3	126.6
Range (cm)	36	26	58	43	3	34
Uncertainty in measure (cm)	18.0	13.0	29.0	21.5	1.5	17.0
Uncertainty in mean (cm)	10.4	7.1	16.7	12.4	0.8	9.8
Percentage error (%)	78.7	30.5	60.7	32.7	20.5	26.7
Percentage decrease (%)	-	-	-	58.5	32.8	56.0

Table 6. Displacement in the space measured with the smartphone.

	Acceleration Integral			With Kalman Filter		
Real distance (cm)	50	70	100	50	70	100
Mean (cm)	76.3	90.6	162.6	57.0	80.6	151.6
Range (cm)	20	28	53	17	25	65
Uncertainty in measure (cm)	10.0	14.0	26.5	8.5	12.5	32.5
Uncertainty in mean (cm)	5.7	8.0	15.3	4.9	7.2	18.7
Percentage error (%)	52.7	29.5	62.7	16.7	20.0	51.7
Percentage decrease (%)	-	-	-	68.3	32.3	17.5

6. Computation Time

The overhead introduced by the Kalman filter was quantified by running an extensive set of computing tasks. In detail, the data acquired from the accelerometer after a movement was processed 64,000 times repeatedly. The current time was recorded at the beginning and at the end of the loop. The time required to execute the loop was computed as the difference between the two moments, which is reported in Table 7. With the Waspmote, the estimation of the displacement by means of the Kalman filter required almost the double the time spent in simply integrating the acceleration data. With the smartphone, the time needed by the Kalman filter increased by almost 50%. However, it can be noted that the time spent for a single estimation was only a few microseconds and, so, the overhead due to the Kalman filter can be considered negligible in almost all the cases, even for time-critical applications.

Table 7. Computation time with Waspmote and smartphone.

	Acceleration Integral	**With Kalman Filter**
Waspmote	1.037 s	2.023 s
Smartphone	0.200 s	0.298 s

7. Conclusions

The estimation of displacement based on two-fold integration of data acquired from an accelerometer is impractical, in most cases. In fact, the estimation tends to be much higher than the real case, and the percentage error between the two values can be notable. A significantly better estimation of the displacement can be obtained by exploiting a Kalman filter for data processing. By running a set of experiments with two different commercial accelerometers, it was observed that the average value of the estimations came closer to the real displacement, along with a lower uncertainty in the measure. The percentage error made in the estimations dropped by up to 72%. This technique, based on the Kalman filter, has thus been shown to be beneficial and can be always suggested, with respect to the integration of acceleration data, as the overhead introduced is not significant. As a future work, the Kalman filter will be tested in combination with other algorithms.

References

1. Jiang, Y.; Zhang, L.; Wang, L. Wireless sensor networks and the Internet of Things. *Int. J. Distrib. Sens. Netw.* **2013**, *2013*. [CrossRef]
2. Shieh, J.; Huber, J.E.; Fleck, N.A.; Ashby, M.F. The selection of sensors. *Prog. Mater. Sci.* **2001**, *46*, 461–504. [CrossRef]
3. Chen, K.Y.; Bassett, D.R.J. The technology of accelerometry-based activity monitors: Current and future. *Med. Sci. Sports Exerc.* **2005**, *37*, 490–500. [CrossRef] [PubMed]
4. Pedley, M. Tilt sensing using a three-axis accelerometer. *Free Semicond. Appl. Note* **2013**, *1*, 2012–2013.
5. Li, X.; Xu, Q.; Li, B.; Song, X. A Highly Reliable and Cost-Efficient Multi-Sensor System for Land Vehicle Positioning. *Sensors* **2016**, *16*, 755. [CrossRef] [PubMed]
6. Mathie, M.J.; Coster, A.C.F.; Lovell, N.H.; Celler, B.G. Accelerometry: Providing an integrated, practical method for long-term, ambulatory monitoring of human movement. *Physiol. Meas.* **2004**, *25*, R1. [CrossRef] [PubMed]

7. Yang, C.C.; Hsu, Y.L. A review of accelerometry-based wearable motion detectors for physical activity monitoring. *Sensors* **2010**, *10*, 7772–7788. [CrossRef] [PubMed]

8. Saunders, J.B.; Inman, V.T.; Eberhart, H.D. The major determinants in normal and pathological gait. *J. Bone Jt. Surg.* **1953**, *35*, 543–558. [CrossRef]

9. Morris, J.R.W. Accelerometry—A technique for the measurement of human body movements. *J. Biomech.* **1973**, *6*, 729–736. [CrossRef]

10. Roylance, L.M.; Angell, J.B. A batch-fabricated silicon accelerometer. *Trans. Electron Devices* **1979**, *26*, 1911–1917. [CrossRef]

11. Szydło, T.; Konieczny, M. Mobile and wearable devices in an open and universal system for remote patient monitoring. *Microprocess. Microsyst.* **2016**, *46*, 44–54. [CrossRef]

12. Culhane, K.; O'connor, M.; Lyons, D.; Lyons, G. Accelerometers in rehabilitation medicine for older adults. *Age Ageing* **2005**, *34*, 556–560. [CrossRef] [PubMed]

13. Maybeck, P.S. *Stochastic Models, Estimation, and Control*; Academic Press: Cambridge, MA, USA, 1982; Volume 1.

14. Bishop, G.; Welch, G. An introduction to the Kalman filter. *Proc. SIGGRAPH Course* **2001**, *8*, 41.

15. Ferrero, R.; Gandino, F.; Hemmatpour, M.; Montrucchio, B.; Rebaudengo, M. Exploiting accelerometers to estimate displacement. In Proceedings of the 5th Mediterranean Conference on Embedded Computing (MECO), Bar, Montenegro, 12–16 June 2016; pp. 206–210.

16. Seifert, K.; Camacho, O. *Implementing Positioning Algorithms Using Accelerometers*; Application Note AN3397: Freescale Semicond, AZ, USA, February 2007.

17. Yun, X.; Bachmann, E.R.; Moore, H.; Calusdian, J. Self-contained position tracking of human movement using small inertial/magnetic sensor modules. In Proceedings of the International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 2526–2533.

18. Slifka, L.D. An Accelerometer Based Approach to Measuring Displacement of a Vehicle Body. Ph.D. Thesis, Master of Science in Engineering, Department of Electrical and Computer Engineering, University of Michigan, Dearborn, MI, USA, 2004.

19. Smyth, A.; Wu, M. Multi-rate Kalman filtering for the data fusion of displacement and acceleration response measurements in dynamic system monitoring. *Mech. Syst. Signal Process.* **2007**, *21*, 706–723. [CrossRef]

20. Han, S. Retrieving the time history of displacement from measured acceleration signal. *KSME Int. J.* **2003**, *17*, 197–206. [CrossRef]

21. Han, S. Measuring displacement signal with an accelerometer. *J. Mech. Sci. Technol.* **2010**, *24*, 1329–1335. [CrossRef]

22. Thong, Y.K.; Woolfson, M.S.; Crowe, J.A.; Hayes-Gill, B.R.; Challis, R.E. Dependence of inertial measurements of distance on accelerometer noise. *Meas. Sci. Technol.* **2002**, *13*, 1163–1172. [CrossRef]

23. Thong, Y.K.; Woolfson, M.S.; Crowe, J.A.; Hayes-Gill, B.R.; Jones, D.A. Numerical double integration of acceleration measurements in noise. *Measurement* **2004**, *36*, 73–92. [CrossRef]

24. Microelectronics, S. LIS344ALH. Available online: https://www.st.com/resource/en/datasheet/lis344alh.pdf (accessed on 8 April 2019).

25. Microelectronics, S. LIS331DLH. Available online: https://www.st.com/resource/en/datasheet/lis331dlh.pdf (accessed on 8 April 2019).

26. Libelium. Available online: www.libelium.com/products/waspmote (accessed on 8 April 2019).